R&S®DDF
Antenna Error Correction
Manual

4074.0968.02 — 01

Radiomonitoring & Radiolocation

Manual

ROHDE&SCHWARZ

Muehldorfstr. 15, 81671 Munich, Germany
Phone: +49 89 41 29 - 0
Fax: +49 89 41 29 12 164
E-mail: info@rohde-schwarz.com
Internet: http://www.rohde-schwarz.com

The following abbreviations are used throughout this manual: R&S® is abbreviated as R&S.

Contents

1 General

Antenna error correction is capable to eliminate some linear distortions caused by physical insufficiencies of the components involved in the DF process (not the direction finder itself) in the context of high frequency effects, antenna shortcomings, cable attenuation and so on. Correction is performed the way that correction data sets are either known in advance (e.g. for a certain antenna) or have to be collected in a prior measurement process, afterwards are deposited in the DF unit and then can be applied to each subsequent bearing result.

Correction data is used to compensate for these factors which have their origins outside the unit itself, but affect the unit's measurement results. The correction data is stored within the flash file system of the DF unit.

Correction data can come into being by several means:

- Delivery by the manufacturer (in case of antenna factors, see chapter 3.2.1, "Antenna Factors", on page 17),
- Measurement of surroundings and manual calculation of appropriate correction (e.g. with cable attenuation, see chapter 3.2.2, "Cable Attenuation", on page 17, or omniphase correction, see chapter 3.2.4, "Omniphase Correction", on page 18),
- Measurement of surroundings and automatic calculation of appropriate correction (e.g. with antenna surroundings, i.e. azimuth correction data, see chapter 3.2.3, "Azimuth Correction", on page 17).

From that, four types of correction data files exist in parallel:

- Antenna factors,
- Cable attenuation,
- Azimuth correction,
- Omniphase.

An overview of the way the correction data reach the DF unit is shown in figure 1-1.

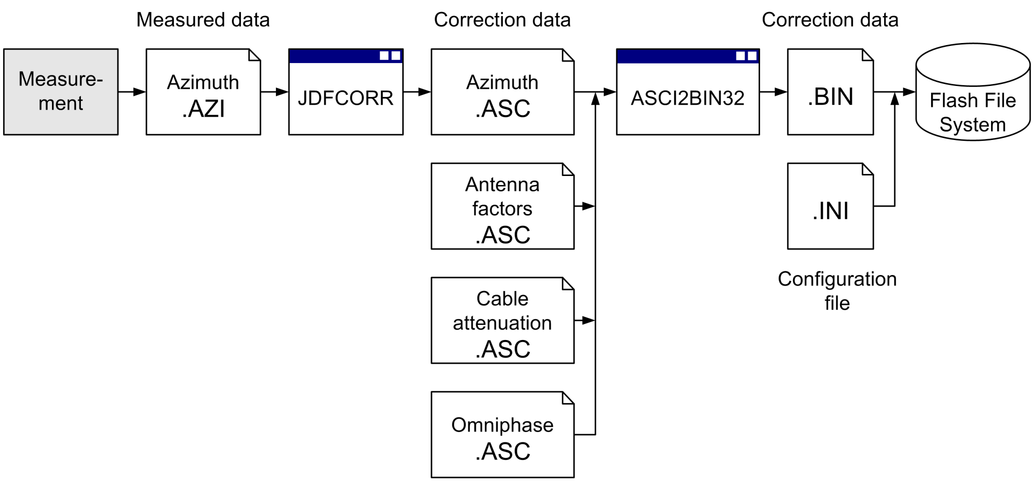

Fig. 1-1: Block diagram correction data.

2 JDFCORR

2.1 General

Denotation

- **Nominal azimuth**

denotes the **real** (geometric) angle the emitter is situated in (as seen from the position of the direction finder), whereas

- **actual azimuth**

means the azimuth value **detected** by the direction finder; this value can be influenced (distorted) by multiple impacts some of which are corrected (eliminated) as described subsequently.

Azimuth correction data are the data needed to convert

- azimuth bearing values measured during later DF (direction finding) operation (subsequently called **actual** azimuth values)

to

- true azimuth values (the azimuth direction the emitter is located in in reality, subsequently called **nominal** azimuth values)

as precisely as possible.

The method to obtain these correction data is to perform a measuring procedure in the "disturbed" environment (see product or antenna publications for detailed informations), covering all azimuth and frequency ranges the later DF operation has to take place in. From the measured data produced thereby the correction data can be derived (calculated).

This calculation of correction data from the azimuth values yielded by the measuring procedure (deviating from the true, i.e. expected azimuth data) is done by the JDFCORR tool which is described in the following chapters.

2.2 Requirements

To run the JDFCORR (**J**ava **D**irection **F**inder **CORR**ection data calculation) tool, you need the

- Java SE Runtime Environment

installed on your computer. For convenience, you find a copy of this environment together with the software shipped; for correct functioning, it is recommended to exclusively use this version (and not separately existing versions). For additional information about Java, visit http://www.oracle.com.

Version of Java SE Runtime Environment

Never use a version of the Java Environment aberrant from the copy shipped with the software; correct functioning cannot be granted in this case.

As can also be seen from the `readme.txt` file available, the remaining requirements are:

- the runnable Java archive `JDFCorr.jar` itself,
- the configuration parameters `Config.xml` and
- the format conversion tool `asci2bin32.exe` (which will be run when storing the correction data, see chapter 2.9, "Writing Correction Data", on page 13)

have to be in the same directory (figure 2-1). Again, you find a default `Config.xml` file together with the software shipped.

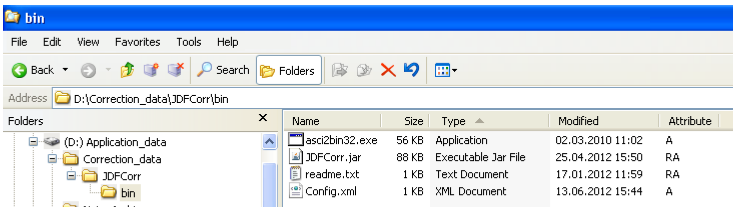

Fig. 2-1: JDFCorr, file directory.

For storing the final correction data in the flash file system of the direction finder (see chapter 5, "Flash File System", on page 29) you will need a usual

- FTP tool,

and for performing manual modifications in the data files a standard

- ASCII editor.

You have performed a measuring procedure as described under chapter 2.1 and have stored all of the measured azimuth data in a plain text (human readable) file. This file is not subject to any constraints like as length etc, rather the data also may be divided into several files, for example, one file per azimuth value (resulting from the course of the measuring procedure: position DF equipment or signal generator to the next azimuth and pass through the entire frequency range). The file extension of all of the files, however, has to be '.AZI'; and, additionally, all matching files have to be stored in a single directory.

2.3 Data Format

For the format of the text file a linewise arrangement is to be kept: one data set per line. Each line follows the format

```
<ValidFlag>; <Frequency>; <NominalAzimuth>; <MeasuredAzimuth>
```

this is, it consists of four columns, the entries of which are separated by the separator character `';'`. For better readability, blanks may exist after the separator signs.

Comments (or comment lines) start with a '#' character and last to the end of a line. Comments, as usual, are excluded from any further evaluation (will not be taken into account).

Example:

```
# Measurement for vehicle y, antenna R&S ADDx
s; 30000000; 10; 10.5
s; 31000000; 10; 11.1
s; 32000000; 10; 11.0
s; 30000000; 20; 20.5
s; 31000000; 20; 21.5
```

- Column 1: **Valid Flag**.
  The Valid Flag labels the measured data of the current line valid or invalid. The letter 's' (meaning "signal") shows valid data; any other character (e.g. 'n' for "noise") appearing here prevents the data of the line from being evaluated.
- Column 2: **Frequency**.
  The value of the (measured) frequency in Hz. If more than one line labeled valid by the Valid Flag have the same frequency and nominal azimuth (column 3) specified, the mean value of the measured azimuth values (column 4) will be calculated and used for evaluation.
  In line 2 (first valid line) of the example: 30 MHz.
- Column 3: **Nominal Azimuth**.
  The expected azimuth value, i.e., the azimuth direction the transmitted wave incides from, in degrees (°).
  In line 2 of the example: 10°.
- Column 4: **Actual Azimuth**.
  The azimuth value found by the direction finder in reality, in degrees (°).
  In line 2 of the example: 10.5°.

2.4 Starting JDFCORR

You start JDFCORR by simply executing (double-clicking) JDFCORR.jar. Of course you can also create an alias on your desktop, in the task bar or the start menu.

The main window of JDFCORR (figure 2-2) welcomes you.

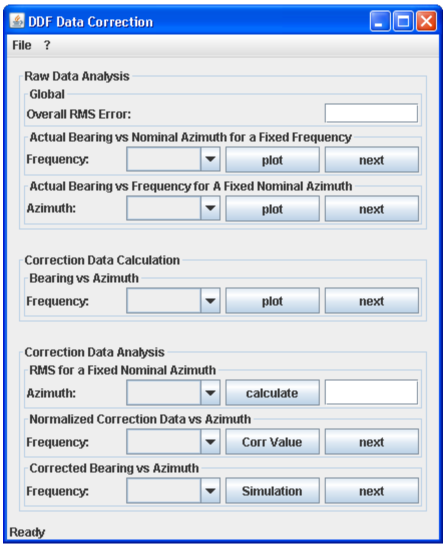

Fig. 2-2: JDFCORR, main window, no data file(s) opened.

2.5 Opening the Input Data

The first thing to do is to make the measured data, therewith the input data, available to the tool. You do this by selecting the item **Open Input Data** from the **File** menu (figure 2-3); the **Select Working Folder** window (figure 2-4) will pop up and let you specify the folder the input file (or files) can be found in. You select the correct folder as usual by either navigating through the folder tree within the **Search In** drop-down menu or typing the correct folder name directly into the **Folder Name** field.

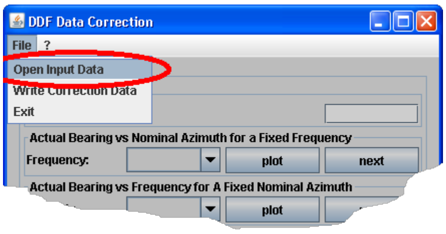

Fig. 2-3: JDFCORR, Open Input Data item in File menu.

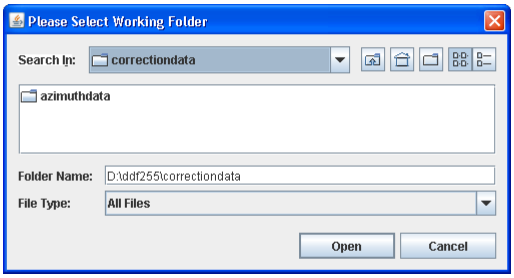

Fig. 2-4: JDFCORR, Select Working Folder window.

Be aware that you have to tell a folder name, but not a file name, since always all files available in the folder will be used for evaluation.

After the **Open** button has been pressed and the data is loaded, most of the input fields propose default values and the count of files opened is shown in the lowermost line of the window (figure 2-5).

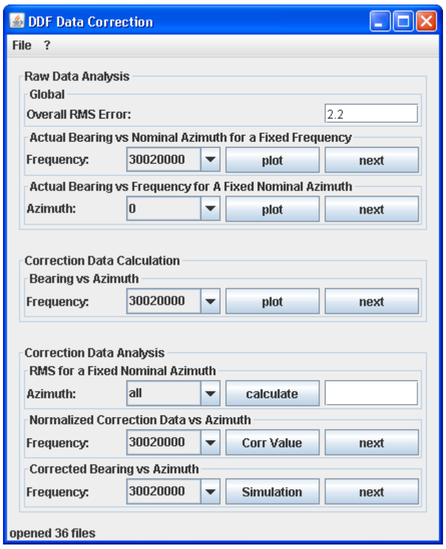

Fig. 2-5: JDFCORR, main window, data file set opened.

The JDFCORR main window offers three sections:

- **Raw Data Analysis**
- **Correction Data Calculation**
- **Correction Data Analysis**

2.6 Monitoring and Displaying the Input Data

The first, the **Raw Data Analysis** section, which deals with visualizing the existing measured data (no correction data calculated so far), on its part consists of three fields:

- **Global**
  This field calculates the **Overall** RMS **Error**. This error is given by

$$\sqrt{\frac{\sum_{n}\left(bearing_{\text{actual}} - azimuth_{\text{nominal}}\right)^2}{n}}$$

  and calculated over all bearing (actual) / azimuth (nominal) value pairs of all frequencies in all data files supplied with the directory specified. Calculation is done automatically when opening a file set (specifying a valid directory).
- **Actual Bearing vs Nominal Azimuth for a Fixed Frequency**
  For the specified frequency, the actual azimuth values are plotted vs. the nominal azimuth values after operation of the **Plot** button. Select another frequency in the

Frequency drop-down menu if desired, or, if you want to obtain a plot for the next (higher) frequency, just press the **Next** button.
A window (figure 2-6) pops up for this purpose; you are allowed to change its dimensions according to your needs (these dimensions are valid also to future plot windows opening). The maximum count of plot windows remaining open, however, is 10; if you cause more plots to be produced, the oldest will be closed.

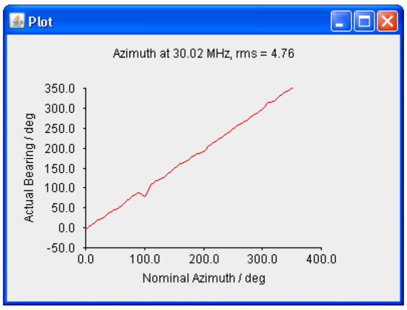

Fig. 2-6: JDFCORR, plot of actual azimuth (actual bearing) vs. nominal azimuth.

- **Actual Bearing vs Frequency for a Fixed Nominal Azimuth**
  For observing the frequency dependence of the azimuth deviation (actual from nominal), this deviation (not the azimuth itself) may be plotted here vs. frequency (figure 2-7); again, the RMS deviation is additionally told in the graph. The azimuth this is to be done for is specified in the according drop-down menu; the steps offered depend on the azimuth steps in the original file (with the measured data). For every nominal angle originally measured the drop-down menu will present an entry (for example: every 10°).

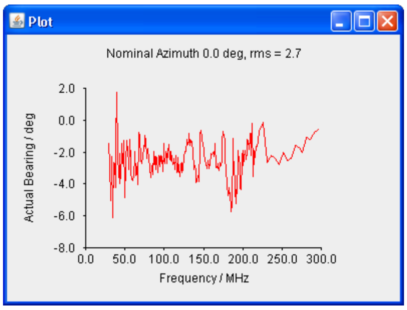

Fig. 2-7: JDFCORR, plot of deviation of actual azimuth (actual bearing) vs. frequency.

2.7 Monitoring Correction Data

The second, the **Correction Data Calculation** section is intended for analyzing the correction data, especially in difficult situations. The only field this section consists of is

- **Bearing vs Azimuth**
  Again for the selected frequency (the next one by the **Next** button), a calculation of correction data can be initiated by operating the **Plot** button. The first step to take for this purpose is to determine whether there are areas (or ranges) where a correction is not possible in principle (see chapter 2.11 for a more detailed explanation). These areas can be recognized by the negative slope in the diagram of actual vs. nominal

azimuth (100° in figure 2-6); they have to be smoothed out by eliminating the measured values causing them and connecting the two respective adjacent measured values directly.
Together with this procedure, a "Fit graph" is plotted (figure 2-8); it again plots bearing vs. azimuth values and thus gives an idea of the degree up to which a correction can be done at all. With the exception of the areas told, the curve resulting is the same as in the "plot bearing vs. azimuth" case.

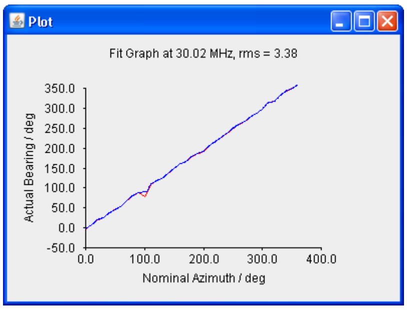

Fig. 2-8: JDFCORR, plot of actual azimuth (actual bearing) and corrected bearing vs. nominal azimuth.

Red line = (cf figure 2-6)
Blue line = Fit graph

2.8 Calculating Correction Data

The third, the **Correction Data Analysis** window consists of three fields:

- **RMS for a Fixed Nominal Azimuth**
  This section allows to calculate an RMS value over just one azimuth direction but all frequencies documented in the opened files. Alternatively, by selecting the **all** item from the drop-down menu, the RMS value can be determined for all values at all, similarly to the value in the **Global** item in the **Raw data analysis** field (chapter 2.6).
- **Normalized Correction Data vs Azimuth**
  By operating the **Corr Value** button, you obtain a plot (figure 2-9) of the correction value (amount of correction need) in ° vs. the actual bearing.
  Again, this is valid for the frequency specified in the **Frequency** drop-down menu to the left, choosing just the next one can also be done by the **Next** button.

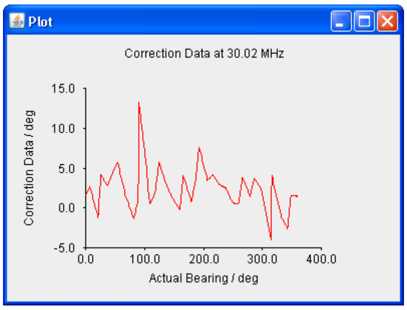

Fig. 2-9: JDFCORR, plot of azimuth correction values vs. actual azimuth (actual bearing).

- **Corrected Bearing vs Azimuth**

In conclusion, this field confirms whether a perfect correction of the measured azimuth deviations could be performed or not (figure 2-10). Values that could not be corrected appear evidently (the spike at 100° in the figure).

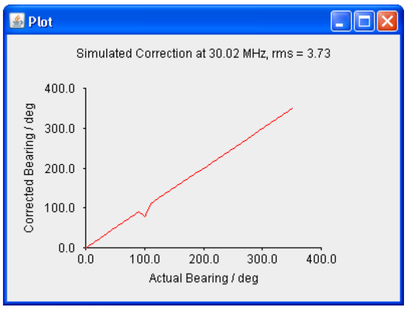

Fig. 2-10: JDFCORR, plot of corrected azimuth vs. actual azimuth (actual bearing).

2.9 Writing Correction Data

Instead of calculating correction data for each frequency separately as described with chapter 2.7 (which of course would be very tedious), you generate all correction data (i.e., the data for all fixed points – azimuth values and frequencies – in all files in the specified directory) simply by operating the **Write Correction Data** item in the **File** menu (figure 2-11).

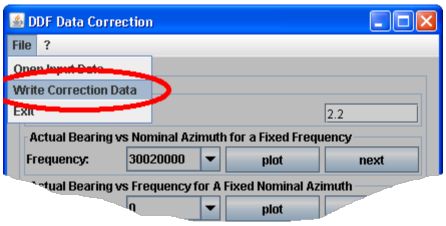

Fig. 2-11: JDFCORR, Write Correction Data item in File menu.

After having done so, a configuration window (figure 2-12) will open and prompt you to enter some heading information for identifying the file uniquely later on – the most important of which is the name of the file to be created. Do not specify a file extension (file type), this will be supplemented automatically, just tell a plain name.

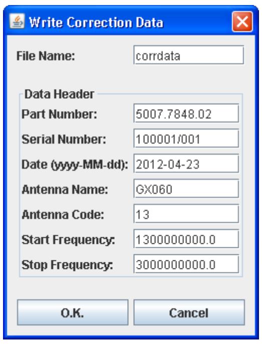

Fig. 2-12: JDFCORR, Write Correction Data Config window.

The result will be two files, one with the extension '.ASC' containing all correction data in the mandatory format (see chapter 3, "Correction Data File", on page 16), the other one in a '.BIN' format.

A message "output file written" appears in the lowermost line of the main window.

If the date entered in the corresponding field is incorrect, an error message (figure 2-13, left) will occur, as is if conversion from '.ASC' to '.BIN' has failed (figure 2-13, right).

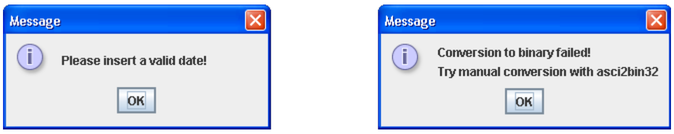

Fig. 2-13: JDFCORR, Wrong Date window and Conversion Failed window.

The '.ASC' format is human readable, i.e., the data header information of the above-mentioned configuration window will appear in the file in plain text and therefore can be altered or supplemented later at any time. The '.BIN' file, in contrast, can be loaded to the flash file system of the direction finder directly, but (e.g. if found to be incorrect) cannot be changed by the user. If changes to a '.BIN' file appear to be needed, repeat the steps described above, or, alternatively, change the '.ASC' file accordingly and convert it to '.BIN' format (see chapter 4, "Conversion to Binary File", on page 28).

2.10 Closing JDFCORR

JDFCORR is terminated with the **Exit** item in the **File** menu (figure 2-14), or by the common WINDOWS methods (clicking onto the top right closing symbol or operating the <ALT><F4> key combination).

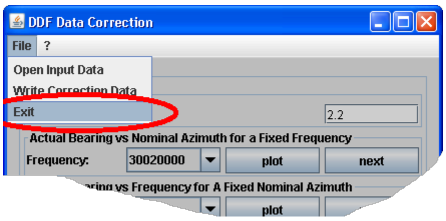

Fig. 2-14: JDFCORR, Terminate.

2.11 Limitations to Azimuth Correction

In the plots in the preceding chapters, a case of limitation to the correction of azimuth has been shown. A (unambiguous) correction is not possible if the measured azimuth values are not monotonically increasing (for increasing nominal azimuth values), this is, if the slope of the function of figure 2-6 turns negative.

An example will show (table 2-1):

Table 2-1: Example: actual (measured) azimuth not increasing.

Azimuth	nominal	0°	10°		20°		30°		40°	50°
	actual	4.5°	15.5°	?	24.5°	?	20.5°	?	42.5°	53.5°
	operational			**21°**		**21°**		**21°**		

The actual (measured) azimuth is not increasing in relation to the nominal azimuth (slope < 1, because measured 20.5° belonging to nominal 30° is less than measured 24.5° belonging to nominal 20°). So, if the direction finder later on finds measured 21°, there is no possibility to decide where to arrange the value: between (nominal) 10° and 20°, between 20° and 30° or between 30° and 40°, i.e., an interpolation cannot be performed in principle.

From that, if negative slopes have been found during measuring cycle, the corresponding value has to be excluded from further consideration and the two adjacent values connected directly as shown in the plot graphs.

3 Correction Data File

3.1 General

Files that contain correction data have the file type (extension) '.ASC' and consist of plain (human readable) text. Correction data are data being used directly for correction of actual azimuth (bearing) data; they are not to be confused with measured data (file type '.AZI') that are described in chapter 2, "JDFCORR", on page 6.

As explained in chapter 1, "General", on page 5, four types of correction data files exist in parallel:

- Antenna factors,
- Cable attenuation,
- Azimuth correction,
- Omniphase.

3.2 Data Formats

As mentioned above, an '.ASC' file contains plain text. It is divided into several blocks belonging to different block types.

In every file (of any block type) have to exist:

- the identification block (always the first block of a file): `IDENTIFICATION_BLOCK`,
- the antenna identification block (second block if a file, telling the type and properties of the antenna connected): `USERBLK_ANTENNA_IDENT`,
- the end block (last block of a file): `END_BLOCK`.

Additionally blocks exist being typical for the appropriate file type, they are described in the corresponding chapter.

Each block contains data records made up of a keyword (terminated by ':') and additional value(s) (also described on the spot) – by reason that not least the order of the records in the block is preassigned, a diligent handling of the data is to be paid attention to.

Again, comments that are excluded from any further evaluation, are to be introduced by '#', they are valid for the remaining line (end with the end of the line); be aware that comments following keywords as 'ALIGNMENT:', 'FORMAT:' etc are not permitted.

Maximum length of a line is 76 characters. The maximum count of nodes that can be specified per antenna is 1000 for the azimuth data, 500 for all other data.

3.2.1 Antenna Factors

A file with antenna factor correction data consists of the above-mentioned mandatory blocks (chapter 3.2) and additional antenna correction data blocks introduced by `USERBLK_ANT_FACTOR_RXDF`.

If an antenna can be brought to multiple configurations, each possible configuration is to be described in a distinct block so that several data blocks arise. Possible configurations of an antenna are:

- Polarization
  - linear: vertical or horizontal,
  - circular: left-hand (counter-clockwise) or right-hand (clockwise),
- Preamplifier: active or passive,
- Mode: DF or receiving antenna.

Therewith, the file structure is

- `IDENTIFICATION_BLOCK`
- `USERBLK_ANTENNA_IDENT` (antenna type)
- `USERBLK_ANT_FACTOR_RXDF` 1 (correction data k-factors antenna configuration 1)
- `USERBLK_ANT_FACTOR_RXDF` 2 (correction data k-factors antenna configuration 2)
- ...
- `USERBLK_ANT_FACTOR_RXDF` n (correction data k-factors antenna configuration n)
- `END_BLOCK`.

3.2.2 Cable Attenuation

A file with cable attenuation correction data consists of the above-mentioned mandatory blocks (chapter 3.2) and an additional block with cable attenuation correction data introduced by `USERBLK_CABLE_ATTENUATION`.

Therewith, the file structure is

- `IDENTIFICATION_BLOCK`
- `USERBLK_ANTENNA_IDENT` (antenna type)
- `USERBLK_CABLE_ATTENUATION` (correction data cable attenuation)
- `END_BLOCK`.

3.2.3 Azimuth Correction

A file with azimuth correction data consists of the above-mentioned mandatory blocks (chapter 3.2) and additional blocks with azimuth correction data introduced by `USERBLK_AZIMUT`.

Thereby, for each frequency described exactly one distinct data block is provided; the minimum count of blocks is two: one for the origin of the frequency range to correct, one for its conclusion.

Therewith, the file structure is

- `IDENTIFICATION_BLOCK`
- `USERBLK_ANTENNA_IDENT` (antenna type)
- `USERBLK_AZIMUT` frequency 1 (correction data azimuth frequency 1)
- `USERBLK_AZIMUT` frequency 2 (correction data azimuth frequency 2)
- ...
- `USERBLK_AZIMUT` frequency n (correction data azimuth frequency n)
- `END_BLOCK`.

Again, different antenna configurations are to be managed separately.

3.2.4 Omniphase Correction

A file with omniphase correction data consists of the above-mentioned mandatory blocks (chapter 3.2) and additional blocks with omniphase correction data introduced by `USERBLK_OMNI_PHASE`.As described above, several blocks are needed if an antenna can be configured differently.

Thereby, for each frequency described exactly one distinct data block is provided; the minimum count of blocks is two: one for the origin of the frequency range to correct, one for its conclusion.

Therewith, the file structure is

- `IDENTIFICATION_BLOCK`
- `USERBLK_ANTENNA_IDENT` (antenna type)
- `USERBLK_OMNI_PHASE` 1 (correction data omniphase antenna configuration 1)
- `USERBLK_OMNI_PHASE` 2 (correction data omniphase antenna configuration 2)
- ...
- `USERBLK_OMNI_PHASE` n (correction data omniphase antenna configuration n)
- `END_BLOCK`.

3.3 Block Formats

The type of a data block is recognized by its ID which is to be specified as the first data set (introduced by the keyword `ID:`).

3.3.1 General Blocks

3.3.1.1 Identification Block

The Identification Block (ID `IDENT_2`) marks the begin of all data blocks, thus, the start of the file. It contains superordinate data for the file to be able to be identified unambiguously later on.

The block structure is

- `ID`: identification of block type: `IDENT_2`
- `ALIGNMENT`: bit alignment of data structure
- `EEPROMSIZE`: size of EEPROM
- `PARTNUMBER`: R&S part number of antenna
- `HWCODE`: HW revision for FW
- `PRODUCTINDEX`: official HW revision
- `SN`: serial number of antenna
- `PRODUCTDATE`: production date of antenna
- `READCODE`: internal flag
- `TESTINSTRUCTION`: revision of test instruction for end test
- `NAME`: name of antenna.

Example:

```
# --------------------
# IDENTIFICATION_BLOCK
# --------------------

ID:                 IDENT_2        # unique identification of block type, do not change
ALIGNMENT:          32             # bit alignment of data structure, do not change
EEPROMSIZE:         0              # size of eeprom, not used, do not change
PARTNUMBER:         5007.7848.02   # rohde & schwarz part number of antenna, not used
HWCODE:             1              # HW revision for FW, not used
PRODUCTINDEX:       01.00          # official HW revision, not used
SN:                 100001/001     # serial number of antenna
PRODUCTDATE:        2009-04-01     # production date of antenna, not used
READCODE:           1              # internal flag, do not change
TESTINSTRUCTION:    03.00          # revision of test instruction for end test, not used
NAME:               ZT255          # name of antenna, not used
```

The comment expression "not used" in the example means that the information entered in the corresponding field will not be needed (evaluated) by the direction finder. A diligent (veridic) infilling is strongly recommended, however.

3.3.1.2 Antenna Identification Block

In the Antenna Identification Block (ID `USER_18300`) the antenna the subsequent data belong to is described in more detail.

String variables (knowable by `FORMAT: STRING`, "antenna model", "antenna manufacturer" in the example) must be exactly 19 characters in length. If necessary they must be filled with blanks.

The block structure is

- `ID`: identification of block type: `USER_18300`
- `Antenna code`:
  - Rx antenna: not required
  - DF antenna: see table table 3-1
- `Frequency range`
- `Antenna type`:
  - Rx antenna: 0
  - DF antenna: 1
- `(Reserved for future use)`
- `Antenna model`
- `Antenna manufacturer`

Example:

```
# ---------------------
# USERBLK_ANTENNA_IDENT
# ---------------------

ID:       USER_18300          # unique identification of block type, do not change
# Antenna code
FORMAT:   UINT16              # format flag, do not change
   1                          # antenna code
   2009                       # year of correction data generation
   4                          # month of correction data generation
   1                          # day of correction data generation
# Frequency range
FORMAT:   DOUBLE              # format flag, do not change
   20000000.0                 # begin frequency of correction data
   6000000000.0               # end frequency of correction data
# Antenna type
FORMAT:   UINT16              # format flag, do not change
   1                          # antenna type 0: RX, 1:DF antenna
   0                          # reserved
# Reserved for future use - do not remove
FORMAT:   UINT32              # format flag, do not change
   0                          # reserved
   0                          # reserved
   0                          # reserved
   0                          # reserved
```

```
# Antenna model
FORMAT:   STRING                 # format flag, do not change
   "ZT255             "          # name of antenna model (exactly 19 characters)
# Antenna manufacturer
FORMAT:   STRING                 # format flag, do not change
   "Rohde&Schwarz     "          # name of manufacturer of antenna (exactly 19 characters)
```

Table 3-1: Antenna codes.

Antenna					
model	**code**	**model**	**code**	**model**	**code**
R&S ZT255	1	R&S ADD_170	20	R&S ADD_157	39
(reserved)		R&S ADD_070_2GHZ	21	R&S ADD_197	40
R&S ADD_115	3	R&S ADD_RESERVED1	22	R&S ADD_078SR	41
R&S ADD_010	4	R&S ADD_195	23	R&S ADD_075	42
R&S ADD_011	5	R&S ADD_153	24	R&S ADD_253	43
R&S ADD_155	6	R&S ADD_255	25	R&S ADD_295	44
R&S ADD_150	7	R&S ADD_215	26	R&S ADD_196	45
R&S ADD_050	8	R&S ADD_216	27	R&S GX153_5	46
R&S ZT660	9	R&S ADD_090_S	28	R&S GX153_9	47
R&S ADD_070	10	R&S ADD_090	29	R&S ADD_011_100M	48
R&S ADD_071	11	R&S ADD_090_M	30	R&S ADD_007	49
R&S ADD_190	12	R&S GX090C	31	R&S ADD_175	50
R&S GX060	13	R&S ADD_090_VAR12	32	R&S ADD_011SR_150M	51
R&S GX060P	14	R&S ADD_090_M_VAR14	33	R&S ADD_207	52
R&S GX060D	15	R&S ADD_110	34	R&S ADD_253_AP	53
R&S VKE	16	R&S ADD_153SR	35	R&S ADD_295_AP	54
R&S ADD_012_1	17	R&S ADD_050SR	36	R&S ADD_153SR_2	55
R&S ADD_012_2	18	R&S ADD_011SR	37	R&S ADD_307	56
R&S ADD_119	19	R&S ADD_011SR_100M	38		

3.3.1.3 End Block

The End Block contains no data with the exception of an ID code (ID `END`). It marks the end of the file.

The block structure is

- `ID`: identification of block type: `END`

Example:

```
#----------
# END BLOCK
#----------
ID:       END                # unique identification of block type, do not change
```

3.3.2 Special Blocks

Interpolation

Be aware that a correction is never performed for frequency values outside of the specified range – this is: an extrapolation is not done.

3.3.2.1 Antenna Factors Block

In an Antenna Factors Block (ID `USER_18310`), the antenna factors are defined as separate field strength (dB/m-) values for each frequency. All values not specified explicitly will be derived from the values of the two adjacent frequencies by linear interpolation during operation. Therewith, at least two frequency points are needed: one for the start (left or lower border), the other one for the end (right or upper border) of the frequency range to be corrected.

Within a correction data block, the values are specified as pairs of frequency and antenna factor ([Hz]-[dB/m]).

The block structure is

- `ID`: identification of block type: `USER_18310`
- `Antenna configuration`
- `Number of frequency points`
- `Data pairs (frequency/antenna factor)`

The `Antenna configuration` item comprises:

- Antenna preamplifier: active/passive
  Some antennas have an internal amplifier that affects the antenna factors. The flag indicates the setting of this amplifier the following correction data apply to:
  – 1: amplifier passive,
  – 2: amplifier active,
  – 0: any amplifier setting.

  The flags can – if the data are unmodifiedly valid for several settings – be combined in binary (dual) form.
- Antenna mode: Rx/DF
  DF antennas may also be used for simple Rx operation; depending on the operation mode, the antenna factors may differ. The flag indicates this antenna mode the following correction data apply to:
  – 1: antenna in Rx mode,
  – 2: antenna in DF mode,

- 0: any antenna mode.

  Again, the flags can be combined in binary form.

- Antenna polarization: linear vertical, linear horizontal, circular left, circular right
  Some antennas may be operated in more than one polarization; this also commonly affects the antenna factors. The flag indicates the polarization setting the following correction data apply to.
  - 1: linear vertical polarization,
  - 2: linear horizontal polarization,
  - 4: circular left-hand (counter-clockwise) polarization,
  - 8: circular right-hand (clockwise) polarization,
  - 0: any polarization.

  Again, the flags can be combined in binary form.

Example:

```
# -----------------------
# USERBLK_ANT_FACTOR_RXDF
# -----------------------
# ANTENNA CONFIGURATION:
# PASSIVE, DF, VERT
# -----------------------

ID:       USER_18310          # unique identification of block type, do not change
# Antenna configuration
FORMAT:   UINT8               # format flag, do not change
   1                          # antenna preamplifier: 1: PASSIVE, 2: ACTIVE, 0: any
   2                          # antenna mode: 1: AF, 2: DF, 0: any
   1                          # antenna polarization:
                              #  1: VERT, 2: HOR, 4: LEFT, 8: RIGHT, 0: any
   0                          # reserved
   0                          # reserved
   0                          # reserved
# Number of frequency points
FORMAT:   COUNT16             # format flag, do not change
   28                         # number of frequency points * 2
# Data pairs (frequency/antenna factor)
FORMAT:   DOUBLE              # format flag, do not change
   30000000.0                 # frequency of 1st frequency point [Hz]
   0.0                        # antenna factor of 1st frequency point [dB/m]
   31000000.0                 # frequency of 2nd frequency point [Hz]
   10.0                       # antenna factor of 2nd frequency point [dB/m]
   32000000.0                 # ...
   16.0
   33000000.0
   22.0
   34000000.0
   28.0
   35000000.0
   0.0
```

```
36000000.0
-10.0
37000000.0
-10.0
38000000.0
10.0
39000000.0
0.0
50000000.0                    # frequency of i-th frequency point [Hz]
10.0                          # antenna factor of i-th frequency point [dB/m]
50000001.0                    # frequency of (i+1)-st frequency point [Hz]
0.0                           # antenna factor of (i+1)-st frequency point [dB/m]
1270000000.0
0.0
1280000000.0
20.0
```

The "i-th" and "(i+1)-st" pair in the example represent a changeover point of a switched antenna (border of two ranges of different switching state): always set frequency specifications as close as possible – ideally in, approaching from either side, a distance of 1 Hz (even values of the frequency, e.g. 1300 MHz, always belong to the lower, odd, e.g. 1300.000001 MHz, to the upper range) – to such borders, because interpolation between two frequency settings is always done strictly linearly.

As mentioned above, values specified within each item of "Antenna configuration" may, if the data are unmodifiedly valid for several settings, be combined by simply adding the single values (dual arrangement of bits).

3.3.2.2 Cable Attenuation Block

In a Cable Attenuation Block (ID `USER_18330`) pairs of frequency and cable attenuation ([Hz]-[dB]) are specified, values in between are interpolated linearly during later operation.

The block structure is the same as with the antenna factors:

- `ID`: identification of block type: `USER_18330`
- `Antenna configuration`:
- `Number of frequency points`
- `Data pairs (frequency/cable attenuation)`

As cable attenuation commonly is independent of the antenna preamplifier, mode and polarization, the corresponding flags should be set to 0.

Example:

```
# --------------------------
# USERBLK_ CABLE_ATTENUATION
# --------------------------
# ANTENNA CONFIGURATION:
# ALL, ALL, ALL
# -----------------------
```

```
ID:       USER_18330          # unique identification of block type, do not change
# Antenna configuration
FORMAT:   UINT8               # format flag, do not change
   0                          # antenna preamplifier: 1: PASSIVE, 2: ACTIVE, 0: any
   0                          # antenna mode: 1: AF, 2: DF, 0: any
   0                          # antenna polarization:
                              #  1: VERT, 2: HOR, 4: LEFT, 8: RIGHT, 0: any
   0                          # reserved
   0                          # reserved
   0                          # reserved
# Number of frequency points
FORMAT:   COUNT16             # format flag, do not change
   4                          # number of frequency points * 2
# Data pairs (frequency/cable attenuation)
FORMAT:   DOUBLE              # format flag, do not change
   60000000.0                 # frequency of 1st frequency point [Hz]
   0.0                        # cable attenuation of 1st frequency point [dB]
   70000000.0                 # frequency of 2nd frequency point [Hz]
   10.0                       # attenuation of 2nd frequency point [dB]
```

3.3.2.3 Azimuth Correction Block

With azimuth correction data, as described above, for each frequency a separate Azimuth Correction Block (ID `USER_18340`) is established. Each of these blocks contains the data pairs for the complete azimuth range (0° to 360°); stepping may be different from block to block, but has to be constant (equidistant) within a single block and has always to be 5° or 10°. Count N of azimuth points is specified explicitly, stepping follows therefrom as 360°/N; subsequently the pairs of frequency and azimuth correction value ([Hz]-[°]) follow for each azimuth angle arising.

For intermediate azimuth values, correction values are interpolated linearly during later operation, as are values for intermediate frequency values.

The block structure is

- `ID`: identification of block type: `USER_18340`
- `Frequency point`:
- `Antenna polarization`:
- `Number of azimuth correction values`
- `Azimuth correction values`

Example:

```
# ---------------------
# USERBLK_AZIMUT
# ---------------------
# 15 MHz
# ---------------------
```

```
ID:       USER_18340          # unique identification of block type, do not change
# Frequency point
FORMAT:   DOUBLE              # format flag, do not change
   15000000.0                 # frequency point of azimuth correction [Hz]
FORMAT:   UINT8               # unique identification of block type, do not change
# Antenna polarization
   0                          # antenna polarization:
                              #  1: VERT, 2: HOR, 4: LEFT, 8: RIGHT, 0: any
   0                          # reserved
# Number of azimuth correction values
FORMAT:   COUNT16             # format flag, do not change
   8                          # number of azimuth correction values
# Azimuth correction values
FORMAT:   FLOAT               # format flag, do not change
   0.0                        # new azimuth value [°] for nominal 0°
   55.0                       #  45°
   90.0                       #  90°
   135.0                      #  135°
   180.0                      #  180°
   225.0                      #  225°
   270.0                      #  270°
   315.0                      #  315°

# ---------------------
# USERBLK_AZIMUT
# ---------------------
# 16 MHz
# ---------------------

ID:       USER_18340          # unique identification of block type, do not change
# Frequency point
FORMAT:   DOUBLE              # format flag, do not change
   16000000.0                 # frequency point of azimuth correction [Hz]
# Antenna polarization
FORMAT:   UINT8               # unique identification of block type, do not change
   0                          # antenna polarization:
                              #  1: VERT, 2: HOR, 4: LEFT, 8: RIGHT, 0: any
   0                          # reserved
# Number of azimuth correction values
FORMAT:   COUNT16             # format flag, do not change
   4                          # number of azimuth correction values
# Azimuth correction values
FORMAT:   FLOAT               # format flag, do not change
   350.0                      # new azimuth value [°] for nominal 0°
   80.0                       #  90°
   180.0                      #  180°
   270.0                      #  270°
```

The example shows azimuth steppings (nominal, i.e. expected azimuth) that are equally spaced, but larger than 10°: 45° in the 15-MHz case and 90° in the 16-MHz case. When determining (measuring) the azimuth data, a distance from one azimuth to the next of 10° or even 5° is mandatory, however.

3.3.2.4 Omniphase Correction Block

Similar to the antenna factor correction data, in the Omniphase Correction Block (ID `USER_18320`) data pairs of frequency and azimuth value ([Hz]-[°]) are constructed; for intermediate values a linear interpolation will be done during later operation.

The block structure is

- `ID`: identification of block type: `USER_18320`
- `Antenna configuration`:
- `Number of frequency points`
- `Omniphase correction values`

Example:

```
# -----------------------
# USERBLK_OMNI_PHASE
# -----------------------
# ANTENNA CONFIGURATION:
# ALL, ALL, ALL
# -----------------------

ID:       USER_18320          # unique identification of block type, do not change
# Antenna configuration
FORMAT:   UINT8               # format flag, do not change
   0                          # antenna preamplifier: 1: PASSIVE, 2: ACTIVE, 0: any
   0                          # antenna mode: 1: AF, 2: DF, 0: any
   0                          # antenna polarization:
                              #  1: VERT, 2: HOR, 4: LEFT, 8: RIGHT, 0: any
   0                          # reserved
   0                          # reserved
   0                          # reserved
# Number of frequency points
FORMAT:   COUNT16             # format flag, do not change
   10                         # number of frequency points * 2
# Omniphase correction values
FORMAT:   DOUBLE              # format flag, do not change
   10000000.0                 # frequency of 1st frequency point [Hz]
   0.0                        # omni phase of 1st frequency point [°]
   11000000.0                 # frequency of 2nd frequency point [Hz]
   100.0                      # omni phase of 2nd frequency point [°]
   12000000.0                 # ...
   200.0
   13500000.0
   350.0
   13700000.0
   10.0
```

4 Conversion to Binary File

Prior to being able to save the correction data to the flash file system of the direction finder, they are to be converted from plain text to a binary format. This has either already been done by the JDFCORR tool (figure 2-12 on page 13 in chapter 2, "JDFCORR"), or can be performed retroactively by the command line tool `ASCI2BIN32.EXE`.

Just open a DOS box (figure 4-1) and invoke the tool, specifying additionally the name of the file to be converted. Note that you have to explicitly tell also the extension '.ASC'.

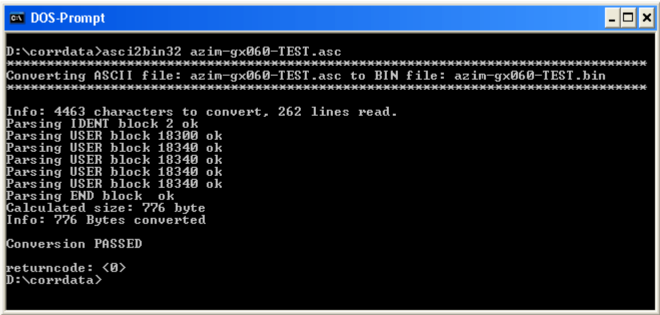

Fig. 4-1: ASCI2BIN32.EXE tool usage.

The file created has the same name as the file specified, but the extension '.BIN'; like all file names, if it later is renamed, the new name must not include blanks; the extension has to be kept '.BIN'.

5 Flash File System

5.1 File Structure

The flash file system has a (invariant) directory structure; it consists of the two directories `\boot` and `\user`. Thereby, `\boot` only contains system files, while user specific files are stored in `\user`. In both directories, the subdirectory `\antenna` exists where the correction data files reside – again subdivided in system immanent data (default data) (directory `\boot\antenna`) and user-specific data (directory `\user\antenna`).

The system immanent data are allocated by Rohde & Schwarz; they comprise standard (default) data for all DF antennas being at disposal. They may be overridden by user-specific data entirely or partly, but they always exist (cannot be deleted), thus, are again available after a possible reboot.

5.2 File Types

5.2.1 Binary Files

Binary files contain correction data in machine readable (binary) form. They are created by the procedure described in chapter 4, "Conversion to Binary File", on page 28 and may exist in any count. Their name is arbitrary (but without blanks contained), but their type (extension) mandatorily has to be '.BIN'.

Example: `k_ADD071.BIN` contains k-factors (antenna factors, also called conversion factors) for the DF antenna R&S ADD071.

5.2.2 Configuration Files

Selection of files to be used (correction data due to the current situation) is done via configuration files, also called CorrSet (Correction Set). These files contain the file names of all correction data (binary) files to be used.

Example:

```
# Test Data for FW V3.00

# Antenna factors
k-ZT255-TEST.bin

# Cable attenuation
cable-ZT255-TEST.bin
```

```
# Omniphase
omni-ADD119-TEST.bin

# Azimuth correction
azim-ADD119-TEST.bin
```

Also in configuration files comments may be inserted that start with '#', last up to the end of the current line and have no influence on further evaluation. Except in file names (see above) blanks may appear at any place and in any quantity.

The name of a configuration file is compulsory as `CorrSet`xy`.INI`, where xy stands for a number of 00 to 99 (CorrSet 0 to CorrSet 99). Selection of a particular configuration file is done by dedicated remote commands, see table table 5-1.

Table 5-1: Remote control commands for configuration file selection.

Remote control standard	Command
SCPI	`SENSe:CORRection:CSET:SELect`
XML	`CorrectionSet`

A special meaning has the configuration file `CorrSet00.INI`: It is situated in the directory `\boot\antenna` – all other configuration files in `\user\antenna` – and contains the names of all default files. Therefore, this name is not permitted to be used for user specific files.

At the moment, `CorrSet00.INI` and the default correction data files the names of which are contained within are not visible from outside (seem to be not present in the directory, but are in reality).

Evaluation of the files is performed the way that the binary files mentioned in the selected configuration file are scanned for their blocks (see above). A block is identified by its ID code, giving the type of the block, and additional specific data as type of the antenna (for which the data are valid), frequency, antenna configuration (setting the data are valid for), etc. A block coinciding in all of these data with a block of the default configuration overrides it until a different antenna configuration is selected (and a system reboot executed).

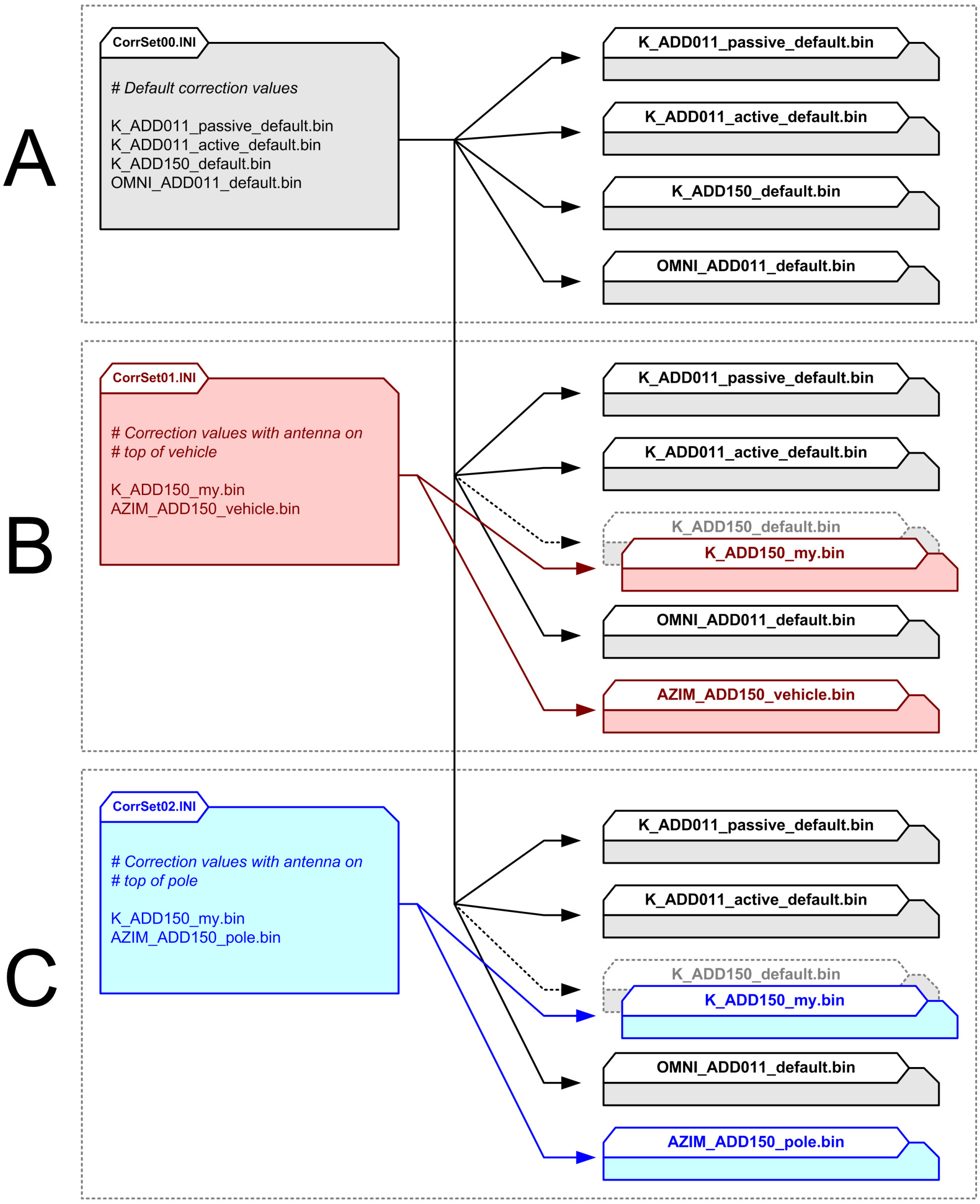

Fig. 5-1: Configuration files.

A = Default configuration file CorrSet00.INI
B = Additional configuration file for vehicle antenna: CorrSet01.INI
C = Additional configuration file for antenna on pole: CorrSet02.INI

An example is shown in figure 5-1:

- Case A shows the situation with no user-defined configuration files: the default file `CorrSet00.INI` houses four antenna (k-factor) data sets. Two
  – `K_ADD011_passive_default.bin` and
  – `K_ADD011_active_default.bin`

  are for two separate configurations (active and passive) of the R&S ADD011, one
  – `K_ADD150_default.bin`

for the R&S ADD150 and the last one
– `OMNI_ADD011_default.bin`

for the omni-antenna connected to the R&S ADD011. No special azimuth correction data are available (as, of course, one would expect with a default configuration).

- Case B adds a configuration file `CorrSet01.INI` describing a vehicle scenario: special antenna factors of the R&S ADD150
  – `K_ADD150_my.bin`

  are provided, therefore overriding the default factors of A (coming from `CorrSet00.INI`), and an azimuth correction file
  – `AZIM_ADD150_vehicle.bin`

  has also been added.
- Case C describes a similar situation, now for a scenario with the antenna mounted on a pole (configuration file `CorrSet02.INI`): again, different antenna factors
  – `K_ADD150_my.bin`

  (i.e., the same as with case B) and azimuth correction data
  – `AZIM_ADD150_pole.bin`

  dominate the default values.

Remember that merely one user configuration file may be active (specified) at a time, being effective in conjunction with the default configuration file `CorrSet00.INI` as shown in the figure.

5.3 Storing

Storing of the prepared data into the flash file system is done by a usual FTP tool (e.g. WinSCP, Filezilla FTP, SmartFTP or various other products). Special attention is to be paid to the fact that the configuration file has to be stored as well as all binary files mentioned within it.

As mentioned above, the destination folder in the file system is `\user\antenna`. Storing can be done from the Windows®command line.

You will be asked for a user name and a password by the FTP tool. Commonly both are the name of the unit itself, but without the leading "R&S", for example `DDF550` for the R&S DDF550. For unambiguity, a complete list is shown in table 5-2.

Table 5-2: User names and passwords for FTP in R&S® units.

Unit	User name	Password	Unit	User name	Password
R&S ESMD	ESMD	ESMD	R&S DDF550	DDF550	DDF550
R&S DDF255	ESMD	ESMD			

Example:

```
ftp <IP address of device>

User name:   <unit name, see above>
```

```
Password:    <unit name>

cd \user\antenna

binary

put CorrSet01.ini
put cable-ZT255-TEST.bin
put k-ZT255-TEST.bin
put omni-ADD119-TEST.bin
put azim-ADD119-TEST.bin
…

bye
```

The correction data having been stored are brought into action by restarting the unit.

Glossary: Abbreviations

Symbols

°: Degree

A

ASCII: American Standard Code for Information Interchange

D

dB: Decibel

DF: Direction Finding

DOS: Disk Operating System, also shortened form of MS-DOS®

E

EEPROM: Electrically Erasable Programmable Read-Only Memory

F

FTP: File Transfer Protocol

FW: Firmware

H

HW: Hardware

Hz: Hertz

I

ID: Identifier

M

m: Meter

MHz: Megahertz

MS-DOS: *Microsoft*® Disk Operating System

R

RMS: Root Mean Square (quadratic mean)

Rx: Receive

S

SCPI: Standard Commands for Programmable Instruments

SE: Standard Edition

X

XML: Extensible Markup Language