MetAir

Applied atmospheric physics and air pollution meteorology
airborne measurements and consulting

corresponding address: Dr. Bruno Neininger, Sonnenberg 27, CH-6313 Menzingen, Switzerland
company registered in CH-8915 Hausen am Albis, Switzerland
e-mail info@metair.ch; URL http://www.metair.ch
fon +41 41 755 38 04, mobile +41 79 340 77 33
fax and voice-box +41 86041 755 38 04

General description of data formats and parameters for all campaigns in 2008:

- **REGAFLUXES (EUFAR, Uni Münster, DE)**
- **FLUXPAT (Uni Köln / FZJ, Bonn/Jülich, DE)**
- **REGAN (EUFAR, Uni Wageningen, NL, flown from Bonn)**
- **ALPRIV (EUFAR, Uni Leeds, UK, flown in Austrian Alps from LSZN)**
  (mainly Laser-Scanning; in-situ data not used until now)
- **TIPPEX (ARC, Tipperary, NT, Australia)**
- **OXK (MPI Jena, flown from Jena and Bayreuth, DE)**

Such a description is an ongoing process, because each campaign, each day of it, and very often each flight needs different aspects to know. Even more different are the foci of the users, their backgrounds and expectations.

Therefore the understanding and the further treatment of the data set during at least one or two years after the campaigns is an iterative process where we need feedback from the users: Questions or requests which are short and not too many at one time. Then I try to react as soon as possible, e.g. during the process of writing a publication.

This document is assumed to be a reference where people are mainly using the electronic version in order to search certain key-words. Nevertheless I try to structure it for other reading as well, and highlight important information.

I guess the most important part is the table of parameters, with the most important remarks included. For some special aspects like "flagging" I add former reports and descriptions in the appendix. Please apologise when such additional information might not be specific for 2008 and/or your campaign.

Thank you for working with our data!

February 2008, Bruno Neininger — Version 1.5

Gaps and parts which need further updates are marked with "$$$"

Find the most actual version as the hidden link www.metair.ch/MetAir_docu_2008.pdf

Complete list of parameters

In case a parameter you find in an output file is not listed here, or you need more information concerning a listed parameter, please send me an e-mail and I will include it as soon as possible! (Check for updates from the hidden link www.metair.ch/MetAir_docu_2008.pdf).

The list below is a reformatted copy/paste from the instruction file which controls the post-processing (a *.jou file). Therefore, also the sequence of parameters is the same. This order might be "unlogical". But, the logic behind is the following: A parameter which is a result of others has to be AFTER the others. Only some special procedures like co-filtering can act backwards: When e.g. a parameter like pFI indicates that LICO2 might be unreliable, then these points in LICO2 are deleted. These and other principles and details are explained in this table (if relevant for the parameter), or thereafter. If you wish you can also access www.metair.ch/MetAir_docu_2008.doc instead where you can sort on your own (e.g. alphabetically) or copy paste into an Excel sheet.

Just one additional important remark before: The sequence of parameters in any output is completely independent. You can choose any subset in any order and reasonable time resolution, and there are many special options. Those that are usually output are marked. As a general rule "everything is possible". So, if you have a need, contact me first. Then I can tell you if it's already implemented, or easy to upgrade, or if you have to care about on your own. The advantage of integrating as many steps of the post-processing in the standard environment metair09.exe is that there are no metafiles and versions. So, if we change something about the temperature and NOx, but, everything remains the same it's just one step of reprocessing. In case you are modifying output files on your own, you have to re-process everything again.

To understand some comments concerning synchronisation of signals it has to be known that in each pod and in the fuselage there is an independent data acquisition system (PC/104, called "cube"). All signals are stored locally (Pod R, Center, and Pod L at 10 Hz), but, most from the pods are also transmitted by 2 Hz to the fuselage (Center) and stored there redundantly. Some of these selected signals marked with a postfix "C" are used to check the synchronicity of the systems (others are just ignored).

For the joint campaign with ARA I have adapted some of the parameter names to the ARA nomenclature. They are reflected in the two rows MetAir and ARA.

MetAir	ARA	unit	comment
			The first 4 parameters marked are calculated when the output is generated. Therefore special rules apply: stime is always first; date and clock can only be generated in averaged files (not in 10-Hz original resolution); FixPnt (if chosen) is always the last row.
stime		[s UTC]	Primary time of the data set. To be accurate, the unit should be "s GPS time". UTC and GPS time is not exactly the same: UTC in 2008 was 14 seconds behind GPS time (12:00:00 h or 43200 s UTC is 12:00:14 h or 43214 s GPS! In 2009 the difference is 1 s more. When defining an output it can be chosen of this time is interval centered (standard), begin or end of interval.
date		[dd.mmy.yyyy]	the date of the flight as day/month/year. Since there were no flights around midnight there are no subtle differences between UTC and GPS. The format is possible with or without "."
clock		[hh:mm:ss]	stime converted to hours, minutes, seconds. The format is possible with or without ":"
FixPnt		[name]	name of the nearest fix point as long as it is within the declared maximum radius (can be different for each individual fix point). Send me the coordinates of any fix point you wish to include. See also frto and prad at the end of this table
ptime		[s]	GPS time of the primary position system (TANS Vector).
vtime		[s]	GPS time of the primary velocity system (TANS Vector).
atime		[s]	GPS time of the primary attitude system (TANS Vector).
long1	Rlon	[°]	longitude from the primary navigation system (MetAir: TANS Vector; ARA: RT4003) in WGS-84 coordinates (Rlon accuracy in the order of 10 cm)
lat1	Rlat	[°]	latitude from the primary navigation system (MetAir: TANS Vector; ARA: RT4003) in WGS-84 coordinates (Rlat accuracy in the order of 10

			cm)
zGPS1	Ralt	[m]	altitude from the primary navigation system (MetAir: TANS Vector; ARA: RT4003) in WGS-84 ellipsoid (Ralt accuracy in the order of 10 cm)
UTMx1		[km]	first order transformation in UTM coordinates from long1/lat1 (compare UTMx2)
UTMy1		[km]	first order transformation in UTM coordinates from long1/lat1 (compare UTMy2)
long2		[°]	longitude from the secondary navigation system (GPS-mouse for the moving map), also in WGS-84. Since 2008 this should be more accurate (about 10 m) than long1 since WAAS/EGNOS is included
lat2		[°]	latitude from the secondary navigation system (GPS-mouse for the moving map), also in WGS-84. Since 2008 this should be more accurate (about 10 m) than lat1 since WAAS/EGNOS is included
UTMx2		[km]	accurate conversion of long2/lat2 into UTM
UTMy2		[km]	accurate conversion of long2/lat2 into UTM
GKx		[km]	accurate conversion of long2/lat2 into "Gauss-Krueger" or other national metric coordinate systems (e.g. Fx for France, CHx for Switzerland)
GKy		[km]	accurate conversion of long2/lat2 into "Gauss-Krueger" or other national metric coordinate systems (e.g. Fy for France, CHy for Switzerland)
zGPS2		[m]	GPS altitude of the backup GPS (WGS-84 ellipsoid, with an accuracy of about 10 m)
edis2	edist	[km]	elapsed distance directly calculated in the moving map software, or by integration (compare with edist)
long		[°]	long1, if there are gaps they are filled with long2
lat		[°]	lat1, if there are gaps they are filled with lat2
UTMx		[km]	UTMx2, if thee are gaps they are filled with UTMx1
UTMy		[km]	UTMy2, if thee are gaps they are filled with UTMy1
zGPS		[m]	zGPS2, if thee are gaps they are filled with zGPS1
vx1	Ru	[m/s]	eastern component of ground speed from primary navigation system
vy1	Rv	[m/s]	northern component of ground speed from primary navigation system
vz1	Rw	[m/s]	upward component of ground speed from primary navigation system
GS1	Rgs	[m/s]	amount of the ground speed from the primary navigation system
TH1	Rtt	[°]	True Track from the primary navigation system (orientation in true north of the GPS/INS track)
GS2		[m/s]	amount of the ground speed from the secondary navigation system
TH2		[°]	True Track from the secondary navigation system (orientation in true north of the GPS/INS track)
azi	Rthdg	[°]	azimuth (= true heading) of the primary navigation system (TANS Vector or RT4003). For each campaign any offset or misalignment with the boom geometry is fine-tuned (typically +/- 0.5 to 2°). The sensitivity for the correction is better than 0.5°.
pit	Rpch	[°]	pitch angle of the primary navigation system (TANS Vector or RT4003). Adjustment see above.
rol	Rrll	[°]	roll angle of the primary navigation system (TANS Vector or RT4003). Adjustment to zero average.
	Racx		longitudinal acceleration from RT4003
	Racy		lateral acceleration from RT4003
	Racz		vertical acceleration from RT4003
faccL	acxRB	[m/s2]	longitudinal acceleration from the sensors in the boom
faccQ	acyRB	[m/s2]	lateral acceleration from the sensors in the boom
faccV	aczRB	[m/s2]	vertical acceleration from the sensors in the boom
td-	zeroML	[V]	zero voltage of the Meteolabor TP3 output for temperature and dewpoint (differential, with common mode)
Tref-		[V]	zero voltage of the Meteolabor TP3 output for the reference temperature (differential, with common mode)
Tref+	trML	[°C]	reference temperature of the Meteolabor TP3 thermocouple and dew point mirror (the thermocouples are measuring against this reference)

	tbRB	[°C]	reference temperature of the Fust probe
utemp		[°C]	the temperature signal from the fast open thermocouple via the voltage signal (noisy)
ftemp	taML	[°C]	the temperature signal from the fast open thermocouple via the frequency signal (V/F conversion) or a low-noise A/D channel (much better than utemp)
	taRF	[°C]	the same from another fine thermocouple (Fust probe) fustRB
tempc		[°C]	the same as utemp, but, from the "cube" in the center, used for synchronisation (see comment in the header)
IRtemp		[°C]	The temperature from the open path IRGA LI-7500, as measured with the original sensor in the longitudinal, perforated housing of the sensor. Together with IRpress the density IRdens in the optical path is calculated.
BoomT		[°C]	The temperature in the boom (in the thermally isolated package of pressure sensors) in order to correct the pressure readings (factory calibration, updated in 2003).
IRpress		[hPa]	The pressure from the open path IRGA LI-7500, as measured with the original sensor in the longitudinal, perforated housing of the sensor (transmitted to the sensor via a 1/8" PFA tube). Together with IRtemp the density IRdens in the optical path is calculated.
p-		[V]	zero voltage of the pressure sensors in the boom (differential measurement)
pc	pxRB	[hPa]	pressure in the central hole of the boom; the offset is checked at least once per campaign against airport QFE or other references
pt	pzRB	[hPa]	pressure difference between the center (pc) and the top hole (pzRB is top-bottom)
pb		[hPa]	pressure difference between the center (pc) and the bottom hole
pl	pyRB	[hPa]	pressure difference between the center (pc) and the left hole (pyRB is right-left)
pr		[hPa]	pressure difference between the center (pc) and the right hole
stau	pqRB	[hPa]	total pressure minus static pressure (german: Staudruck)
	psRB	[hPa]	first order static pressure (for the MetAir boom this is pc minus the average of pt to pr)
press	paRB	[hPa]	(corrected) static pressure. Offset see pc; span is checked via zInt by comparing with zGPS (each flight, also useful for t0ML etc.)
alpha	alphaRB	[°]	angle of attack from the 5-hole probe (multi-parameter optimisation of the calculation). For each campaign alpha and beta is fine-tuned by a factor during the optimisation of wind results (minimum TKE in suitable phases of the flights). The sensitivity for the correction factor is better than one percent.
beta	betaRB	[°]	side slip angle from the 5-hole probe (multi-parameter optimisation of the calculation). Adjustment see above.
alphac		[°]	same as alpha from the central data logger (used for synchronisation and checks)
rdew		[g/kg]	auxiliary (raw) dewpoint used in the next two steps
adew		[g/kg]	difference of rdew to a moving average in order to detect spikes due to radio emissions
txVHF		[g/kg]	detection of radio emissions (VHF voice) in order to filter diverse parameters
dewp	tdML	[°C]	dew point signal from the Meteolabor TP3. When mix-IRH2O is larger than typically 0.5 g/kg (adapted for each flight) these individual dew point readings are deleted (for fast H2O use the fitted IRH2O which normally has no gaps)
e1		[hPa]	first guess of the water vapour pressure (based on the dew point and the non corrected temperature), used for density only where the humidity is only a second order term. The reason for this step is that only with a correct density the kinetic correction of the temperature can be made.

mix1		[g/kg]	first guess of the water vapour mixing ratio (e1)
IRdens		[kg/m3]	density in the open path IRGA LI-7500 (see IRtemp and IRpress)
	uIRH2O	[V]	raw voltage from the LI-7500 for the water channel (VH-EOS only)
	uIRCO2	[V]	raw voltage from the LI-7500 for the CO2 channel (VH-EOS only)
IRH2O		[g/kg]	fast and accurate water vapour mixing ratio from the open path IRGA LI-7500, fitted to mix (dew point mirror) for each flight. This fit is a multi-parameter optimisation using linear regression, drift, second order effects of TAS0 (stau) on IRdens, etc. The accuracy is about 0.2 g/kg.
IRCO2		[ppm]	fast CO2 mixing ratio from the open path IRGA LI-7500, before fitting it to LICO2 (see CO2 below). Since the output from the instrument is a mol-density this primary signal is converted to a mixing ratio (ppm) by using IRdens. The mounting guarantees a homogeneous density in the optical path. Some remaining inhomogeneities are detected and corrected as a function of stau. Also some remaining interferences with IRH2O are detected and corrected – irrespective if this is from calibration errors or dilution. It just guarantees that the average signal is in agreement with LICO2 and the high-accuracy flask results. **Use "CO2" for any scientific work. LICO2 and IRCO2 are only delivered in order to give you an idea about the a-priory correlation of the two.**
aH2O		[g/kg]	difference between mix1 and IRH2O for diagnostics and filtering of dewp and the parameters below based on dewp
e		[hPa]	water vapour pressure (dewp) NOAA formula $$$sc
mix	mixML	[g/kg]	water vapour mixing ratio (press,e) NOAA formula $$$sc
dens1		[kg/m3]	first guess of ambient density(IRtemp,press,e) NOAA formula $$$sc
TAS1		[m/s]	first guess of True Air Speed (stau,dens1)
MLt0	t0ML	[°C]	kinetic correction of the fast temperature measured by the ML3 thermocouple. The recovery factor is optimised from slow ascents / fast descents (minimum hysteresis) and is between 0.5 and 0.8 (different mountings)
	t0RF	[°C]	kinetic correction of the fast temperature measured by the Fust thermocouple. Recovery factor as above
IRt0	t0IR	[°C]	kinetic correction of the temperature measured by the sensor in the optical path of LI-7500 (not relevant for IRdens! – only as a third estimate of ambient temperature). Recovery factor as above
rH		[%]	Relative Humidity (dewp,MLt0)
dens0	densRB	[kg/m3]	final ambient density (MLt0,press,e)
TAS0	tasRB	[m/s]	final True Air Speed (stau,dens0). For each campaign stau is fine-tuned during the optimisation of wind results in order to get best TAS. The sensitivity for the correction factor is better than one permille.
IRpot		[°C]	Potential Temperature based on IRtemp and IRpress
pot		[°C]	Potential Temperature based on MLt0 and press
virt		[°C]	Virtual Temperature (MLt0,mix) mainly used for zInt
zInt		[mMSL]	Integrated pressure/density altitude (press,virt). This altitude is adjusted to zGPS or Ralt in three steps: (i) correct initial QFE; (ii) span checked and adjusted (this is a very useful diagnostics for correct MLt0 – it is an independent and very sensitive absolute calibration for any temperature reading and recovery factor!); (iii) Any synoptic pressure change during the flight (offset at landing) is adjusted by a constant correction of typically +/- 1e-3 m/s.

power		[V]	primary DC power (27 to 29 V when the generator is active; sinking to 24 V when powered from the lead accumulators (26 Ah).
sole	kt15	[W/m2]	pseudo "global radiation" from the Meteolabor photometer (upward looking, 350..950 nm, only semi-quantitative, used as a relative information about general solar radiation (daytime) and clouds. For a more quantitative use the characteristics of the sensor has to be determined and the angle between the sun and the sensor has to be calculated. By doing the latter also the characteristics could be described (envelope of the maximum radiation for each angle).
aer03f		[n/ml]	fast (3 Hz) read out of the optical aerosol counter for particles > 0.3 micrometers. Correction see below.
aer03		[n/ml]	standard read out (1 Hz) of the optical aerosol counter for particles > 0.3 micrometers, corrected for coincident counts since 2006, where we flew this instrument parallel to a more accurate counter.
aer05		[n/ml]	standard read out (1 Hz) of the optical aerosol counter for particles > 0.5 micrometers. Correction as above.
LItemp		[°C]	Temperature in the modified closed path IRGA LI-6262 (measured just behind the outlets of sample and reference cells which are coupled by a temperature controlled heat-exchanger in the inflow). Normally this temperature in the housing is kept around 30°C (40°C in Tippex).
LIpress		[hPa]	Pressure in the modified closed path IRGA LI-6262 (measured in the sample cell which is hydraulically coupled to the reference cell with a short common outlet).
LIe		[hPa]	originally this was the water vapour pressure calculated from LIH2O. However, since we dry the sample air with the cryo-trap and Mg-Perchlorate we set this to zero in order to avoid problems with artefacts and negative signs.
LIdens		[kg/m3]	Density in the optical path(s) of the closed IRGA LI-6262. This is for information only since the conversion from the mol-density to mixing ratio is already made in the instrument (using the same sensors/signals). This is different from IRCO2!
LICO2r		[ppm]	initial (raw) reading from the modified LI-6262. Only an offset correction is applied when known from either the ground calibration or the flasks.
Craw		[mV]	raw voltage from the LI-6262 for the CO2 channel
Hraw		[mV]	raw voltage from the LI-6262 for the water channel
LIH2O		[g/kg]	remaining water vapour mixing ratio in LI-6262 after drying by the cryo-trap and Mg-Perchlorate (the latter not in all flights, then this signal is reporting the performance of the cryo-trap). When negative readings occur in the original data, an offset is applied (typically 0.3 g/kg).
LICO2		[ppm]	final (slow) CO2 concentration from the modified closed path IRGA LI-6262 after applying many filters and corrections which "tie" this parameter to the concentrations found in the flasks. The main filter which causes a lot of gaps is the pressure detector in the tubing used for the filling of the flasks (pFl). By partly unknown effects the LI-6262 reading is disturbed during this process. Then a second order density correction is applied (LIdens seems not to be perfect), and usually also a drift in the order of +/- 1 ppm/h. This slow (about 0.3 Hz) data with gaps where it most interesting (filling of flasks) is combined to CO2 below.
pFl		[hPa]	pressure in the tubing for flask filling, used for the detection and documentation of the fillings.
flask		[-]	indicator that a flask is filled. Based on pFl, used for triggered output (*.xls with all data synchronised to flask fillings)
LICO2c		[ppm]	auxiliary parameter to generate CO2 (time series of LICO2c and IRCO2c with identical gaps)
IRCO2c		[ppm]	auxiliary parameter to generate CO2 (time series of LICO2c and IRCO2c with identical gaps)
LICO2av		[ppm]	moving average (standard is 60 s) of LICO2c
IRCO2av		[ppm]	moving average (standard is 60 s) of IRCO2c
IRoff		[ppm]	difference between LICO2av and IRCO2av used fort the correction of IRCO2

CO2		[ppm]	fast and accurate CO2 mixing ratio from the open path IRGA LI-7500, fitted to LICO2, and hence to the flasks. Therefore this signal is "the best CO2 available": It has the resolution of the LI-7500 (20 Hz downsampled to 10 Hz) and the absolute accuracy of the flasks (<0.1 ppm). After the fit, the residuals are typically <0.5 ppm (see *.xls where this is documented for each flask and hence section of flight).
aH2O		[g/kg]	difference between mix and IRH2O for diagnostics and adjustment
aCO2		[ppm]	difference between LICO2 and IRCO2 for diagnostics and adjustment
UVOr		[V]	raw signal from the ozone photometer (oscillating between scrubbed / non-scrubbed)
scrub		[]	trigger signal for the scrubber valve
TUVO		[°C]	temperature in the single cell ozone photometer
dp		[hPa]	pressure difference ambient-inside in the cell
Uref		[V]	reference voltage (actual sensitivity of the detector)
UVO3		[ppb]	calibrated slow (10 s moving average) ozone mixing ratio
O3q		[ppb]	first guess ozone mixing ratio from the onboard system
PANok		[]	indicator (1/0) that the temperature of the PAN-converter in the NOxTOy is between 115 and 125°C.
NOyOK		[]	indicator (1/0) that the temperature of the NOy-converter in the NOxTOy is between 340 and 360°C.
NO2		[ppb]	NO2 signal from the first channel of the NOxTOy instrument. This is the fastest signal (at least 2 Hz, no converters) which has a line delay of 0.7 s only (corrected for). The calibration is from the last field calibration (very time consuming, therefore not made during each campaign) plus a quantitative plausibility check and final adjustment via the equation [Oxd] = [UVO3]+[NO2]. Under best conditions the absolute accuracy of this signal is 0.5 ppb + 10% of the reading. However, in 2008 the campaigns were not focussed on photochemistry. So better take +/-(1 ppb + 20% of reading), includig the remaining uncertainty of the correction for the ozone interference (2 to 5 %, checked for each campaign, could be refined for each flight). Temporary malfunctions (disturbance of the Luminol film during fast pressure changes) can never be excluded. So check the plausibility of data and discuss with me before drawing any far-reaching conclusions.
NOx		[ppb]	Like NO2, after a CrO3 converter which was checked for efficiency in the May-campaign. The resolution is about 1 Hz, the corrected intake line delay is 1.5 s. The experience is that under summer conditions (enough light) NOx on flight altitudes above 50 mGND is identical to NO2, i.e. [NO] = [NOx]-[NO2] is usually very small (<2 ppb) and hence within the "noise". So, if interested in NO, look for larger differences and discuss the case with me.
NOy		[ppb]	The Molybdenum NOy converter was ok at the beginning of the season, but, deteriorated very quickly. This means that before OXK the NOy has to be corrected by a large factor and has large trailing edges (memory). Sorry, virtually useless. During OXK the signals are plausible and hence we can profit from the NOy which has no interference with O3 (corrected in the other channels). The corrected line delay is 5 s; the temporal resolution an average of about 3 s.
iHNO3		[ppb]	This channel is a second NOy channel where the HNO3 is scrubbed out with a Nylon grid. The temporal behaviour is similar to NOy. For the differences the same remarks as for NO (NOx) apply. For a campaign with useful HNO3 estimates the instrument needs a person exclusively dealing with daily calibrations. However, all these redundant channels are very useful when seeing e.g. an NO2 or NOy peak and asking the question if this was an artefact. Since all 6 channels are completely independent, common peaks are real.
iPAN		[ppb]	NOx+PAN converted to NO2. The small converter (heated PFA tubing) causes a corrected line delay of 2.8 s. [PAN] = [iPAN]-[NOx] is only useful when PAN-concentrations of >2 ppb are present (e.g. Paris 1999). But, the signals could be checked if such phases occur. Then we could improve the calibration for such phases.

Oxd		[ppb]	Oxd is a direct measure of Ox, i.e. O3+NO2, after titration with excessive NO in a dark tube. The volume of the tubing causes a corrected line delay of 9.3 s. As can be seen in most campaigns/flights this signal is reasonable, but, contains some non-filtered spikes and drifts. The drift is not a problem because finally this is corrected for (see Oxf and O3f). For a serious use of these signals we would need a closer look into the relevant phases.
UVOx		[ppb]	calculated Ox as the su of UVO3 and NO2; this can be correlated with Oxd to find out if the measurements are consistent (adjustement of the NO2 calibration after Oxd has been adjusted to UVO3 at higher altitudes (NO2 < 1..2 ppb).
PAN		[]	difference iPAN-NOx which would be PAN if everything is well-calibrated
xNO2		[]	pseudo-NO2 by subtracting UVO3 from Oxd; this can be correlated with NO2 to find out if the measurements are consistent
Oxf		[ppb]	Oxd is fitted to UVOx in 30-s moving averages (similar to tying IRCO2 to LICO2) which gives a more reliable Ox estimate (corrected for any drifts)
O3f		[ppb]	the difference Oxf-NO2 can be used as a fast O3 signal. However, this needs to be checked in phases of flights if you wish to use this data!
COzero		[]	indicator (0/1) that the in-flight zero calibration of the CO monitor is active
COspan		[]	indicator (0/1) that the in-flight span calibration against a reference gas of the CO monitor is active (explanations see below; CO)
TPM		[°C]	temperature of the photo multiplier. This temperature has a secondary influence to the signal. Because we have an in-flight calibration each 10 minutes the dependency can be detected and applied only for the interpolation between the calibrations (very small, not really relevant, you can see the factors applied in the *.jou after "COcali….")
TLamp		[°C]	temperature of the UV lamp (housing where the plasma tube is in). This temperature has a secondary influence to the signal. Because we have an in-flight calibration each 10 minutes the dependency can be detected and applied only for the interpolation between the calibrations (very small, not really relevant, you can see the factors applied in the *.jou after "COcali….."). This signal is sometimes missing (according to the manufacturer a bonding problem in an IC; not worth to repair at the moment).
pN2		[hPa]	Absolute primary pressure of the N2 before the capillary for flushing the optics. Higher pressure means higher flow. This flow has a secondary influence to the signal. Because we have an in-flight calibration each 10 minutes the dependency can be detected and applied only for the interpolation between the calibrations (very small, not really relevant, you can see the factors applied in the *.jou after "COcali…..")
pCell		[hPa]	Documentation of the regulated pressure in the measuring cell. No detectable influence to the signal.
ArFlow		[ml/min]	Flow of the Ar-CO2 mixture ("lamp gas"), regulated by a MFC. Because we have an in-flight calibration each 10 minutes the dependency of the signal can be detected and applied for the interpolation between the calibrations (very small, not really relevant, you can see the factors applied in the *.jou after "COcali…..")
CO		[ppb]	Filtered and calibrated CO concentration. The original temporal resolution is 5 Hz, but, if this resolution is not requested (standard) a moving median filter is applied to remove spikes in a moving 0.5- or 1.0 s interval (check MedAve …. in the COcali line in *.jou). When CO concentrations from flasks are available the final calibration is made by these (similar to LICO2). If not, or redundantly, a known calibration gas is fed to the system after each flight (was mostly done). With this the concentration of the on-board reference gas (small containers which are exchanged and hence unknown after about 5 flights) can be determined. This concentration is visible in the instruction line in *.jou. Since we did both calibration methods in almost all campaigns we can also "re-calibrate" the calibration gas cylinder (tie it to the reference of MPI Jena via the flasks). The CO is a very reliable, fast (1..5 Hz) and accurate (2..4 ppm) signal, but, unfortunately in the environment of FLUXPAT, in the vicinity of the high-power "short wave" radio emitter of Juelich there are many very noisy sequences in the data set. All known EMV measures are already made and it is obious that these disturbances can not be further reduced. It is – in contrary – a "miracle" that not most other parameters are influenced as well, since these emissions are usually even to hear in the headsets (reception via cabling and operational amplifiers of the on board intercom which is also "hardened" since 2003).

COfc		[Hz]	CO raw data transmitted from the LH Pod to the Center; used for the synchronisation of the LH Pod
Ts		[°C]	Surface temperature from the Heimann IR thermometer. The readings are obviously too low. The structure and plausibility has to be checked, e.g. over lakes (marked in the data sets).
vx2		[m/s]	eastward velocity component (GS) from the secondary GPS
vy2		[m/s]	northward velocity component (GS) from the secondary GPS
GSx		[m/s]	Mainly GS1, gaps filled with GS2 (flagged)
edist		[km]	integrated GSx (elapsed distance)
THc		[°]	Mainly TH1, gaps filled with TH2 and then converted to circles instead of degrees (1 circle is 360°, but there are no discontinuities, i.e. circles can add up, LH is negative, RH positive. Depending on the flight pattern and the pilot THc is within +/- 5.
THx		[n]	THc reconverted to degrees
vzp		[m/s]	vertical velocity from pressure difference (like the variometer reading), averaged over 10 s.
acirc		[n]	azi in circles, if gaps occur it is filled with THc
azic		[°]	acirc converted back to degrees
pitc		[°]	mainly pitch, if gaps occur they are filled with an empirical fit (TAS0). For 2008 data this is irrelevant, but there were seasons with major gaps in pitch.
rolc		[°]	Mainly rol, if gaps occur they are filled with an empirical fit (dazi/dt, i.e. curvature). For 2008 data this is irrelevant, but there were seasons with major gaps in pitch.
af		[m/s2]	acceleration transformed longitudinally (along the axis of the fuselage)
ax		[m/s2]	acceleration transformed eastwards
ay		[m/s2]	acceleration transformed northwards
az		[m/s2]	acceleration transformed upwards (independent of rol)
nint		[n]	flight legs numbered. The algorithm is detecting maxima in three-dimensional accelerations, concluding that these are turning points between different flight legs. Can be used to automatically decompose a flight into flight legs.
GSa		[m/s]	In the following a complex procedure is made with the accelerations in the boom, velocities and positions. Basically, the accelerations are integrated to Ground Speeds. These are compared with and corrected to GPS velocities in intervals of typically 20 s. The second integration delivers positions. They are also compared and corrected to GPS position. The algorithm can be run in different modes, e.g. in free run the speed and position is only an integral of three dimensional accelerations. This is very sensitive to misalignments, offsets in accelerations, etc. When all parameters are adjusted, the automatic mode is started where the integrals are tied to the measured velocities and positions. All these complex algorithms are not necessary anymore when using a GPS/INS like the RT3000 as we will use also in MetAir in 2009. However, until now, almost the same performance could be reached with the TANS Vector for attitude, speed and position and the highly sensitive three-axis accelerometers in the boom (synchro-sensors from former "Kistler"). But, an integrated system makes life easier. This first parameter is the integrated Ground Speed (much higher resolution than GS1, GS2 or GSx directly from GPS).
edisa		[km]	This is the integral of GSa, it is the best (detailed) elapsed distance
GSerr		[m/s]	This is an estimate of the error of GS based on residuals (rather pessimistic, i.e. a GSerr of 1 m/s does not mean that the GSa have an

			uncertainty of 1 m/s (RMSD), but, that this is the maximum error)
vxa		[m/s]	same as GSa for eastward direction
xa		[km]	integrated eastern position (usually UTM or another metrical coordinate system chosen)
vxerr		[m/s]	same as GSerr for vxa
vya		[m/s]	same as vxa for northward Ground Speed
ya		[km]	integrated northern position (usually UTM or another metrical coordinate system chosen)
vyerr		[m/s]	same as vxerr, but, for vy
vza		[m/s]	same as vxa or vya for upward speed of the boom
za		[m]	integrated vertical position of the boom (very detailed, with every 10 Hz movement)
vzerr		[m/s]	the error estimate for vza
uverr		[m/s]	the quadratic sum of vxerr and vyerr
uu		[m/s]	In this and the following four lines the flow is calculated as the difference between the velocity vector of the aircraft in the earth-fixed system (GPS and accelerations), and the flow impinging on the boom (5-hole-probe). The result is the three-dimensional wind. uu is the component along the axis of the fuselage (i.e. mailnly GS-TAS)
u		[m/s]	eastward component of the wind
v		[m/s]	northward component of the wind
w		[m/s]	upward component of the wind
TKE		[m2/s2]	Total Kinetic Energy from the three components in 5 or 10 minutes intervals (check the line in *.jou). No factor, i.e. only the sum of squares of the turbulent wind speed components (u',v',w'). This TKE is used for many optimisations, i.e. the hypothesis is that with optimal parameters the TKE is minimised, especially in phases like circling, take-off, change from climb to descent, etc. A procedure "Report" allows to output TKE in specified intervals. Then e.g. the facor for "stau" can be modified and the TKE-report can be minimised.
vel		[m/s]	u,v converted to a amount of velocity (wind speed)
dir		[°]	u,v converted to a wind direction (the direction the wind is blowing from, relative to true north)
Mvel		[m/s]	An alternative algorithm, completely independent of the complex wind calculation for u,v.w is using GS, TH, and TAS only. With a statistical approach (GS-TAS in sectors) a horizontal wind is fitted with a least-squares approach. Even when the attitude angles fail we can get quite decent wind estimates from this algorithm. If all the systems are working this is a redundant information for comparison.
Mdir		[°]	The wind direction from the algorithm explained above.
Merr		[m/s]	The residual (error estimate) from the algorithm explained above
avw		[m/s]	average vertical wind speed in a moving average of 10 minutes (600 s)
dw		[m/s]	the vertical turbulence relative to avw (=w-avw)
avCO2		[ppm]	the average CO2 concentration (moving average of one minute)

avH2O		[g/kg]	the average H2O mixing ratio (moving average of one minute)
avpot		[g/kg]	the average potential temperature (moving average of one minute)
dCO2		[ppm]	the turbulent part CO2' relative to avCO2
dH2O		[g/kg]	the turbulent part H2O' relative to avH2O
dpot		[g/kg]	the turbulent part pot' relative to avpot
MolDens		[mol/m3]	The mol-density dens0 x 34.6
MolHeat		[J/m3/K]	The heat capacity per mol of air dens0 x 1007
CO2flux		[umol*m-2*s-1]	The vertical CO2 flux in moving intervals of 30 s
H2Oflux		[mmol*m-2*s-1]	The vertical H2O flux in moving intervals of 30 s
Eflux		[W*m-2]	The vertical heat flux in moving intervals of 30 s
Avel		[m/s]	mainly vel; gaps filled with Mvel
Adir		[°]	mainly dir; gaps filled with Mdir
mdir		[°]	Adir in mathematical angles (east is 0°, anticlockwise), mainly for graphics with vectors (darts)
DTM	SRTM	[mMSL]	Digital Terrain Height below the flight track from the SRTM data set (Shuttle Radar Topography Mission of NASA)
zGND		[mGND]	height above ground calculated as zInt-DTM
	zLAS	[mGND]	height above ground measured by the laser altimeter
	dtmLAS	[deg]	calculated topography heigt from the precise altitude of the aircraft and the laser altimeter (also crop height and other derived parameters are available). In 2009, also the MetAir-DIMO will have such a laser altimeter (a poor man's version of a laser scanner, "scanning" with 100 Hz only below the flight track)
xSC		[deg]	The eastward coordinate (long or km) of any pointing device like the Hyper Spectral Scanner / Camera on the ground.
ySC		[deg]	The northward coordinate (long or km) of any pointing device like the Hyper Spectral Scanner / Camera on the ground. Also the edges of the images are available.
Ftim		[s]	The time (end of interval) where a flask was filled
F#x		[#]	number of the flask (just numerical – without any pre- or postfix like J- or -10
F#		[#]	F# spread over 30 seconds (not just a point)
X-CO2		[ppm]	the CO2 concentration from the flask at this time/location
F-CO2		[ppm]	X-CO2 spread over 30 seconds (not just a point)
fCO2		[ppm]	the offset of the continuous measurement CO2 from the flask analysis
X-CO		[ppb]	the CO concentration from the flask at this time/location
F-CO		[ppb]	X-CO spread over 30 seconds (not just a point)
fCO		[ppb]	the offset of the continuous measurement CO from the flask analysis
CH4		[ppb]	the Methane concentration from the flask spread over 60 s
N2O		[ppb]	the N2O concentration from the flask spread over 60 s
H2		[ppb]	the Hydrogen concentration from the flask spread over 60 s
SF6		[ppt]	the SF6 concentration from the flask spread over 60 s

d13C	d-13C	[‰]	the depletion of 13C from the flask (ask MPI Jena, Christoph Gerbig or Stepan Baum for the details)
d18O	d-18O	[‰]	the depletion of 18O from the flask (ask MPI Jena, Christoph Gerbig or Stepan Baum for the details)
frto		[from.to]	The flight leg was from point n to point m (n.m, e.g. 5.03 means that the flight leg was from the fifth point in the list to the third point
prad		[km]	The distance to the nearest fix point as long as prad is smaller than a specified maximum radius (individually for each point). You can send me any fix points you are interested in (lat, long) and you can find it in the data set.
GmT		[m/s]	GSa-TAS0 for the diagnostics/optimisation of the wind calculation
pma	pmaRB	[°]	pitc-alpha for the diagnostics/optimisation of the wind calculation

1. Overview (from IsoTrans 2006, needs mods $$$)

1.1 Flight patterns

Specific for each campaign. Ask for maps if not generated on your own.

Table 1: Listing of flights with processed measurements:

Flight	from	to	from	to	durat	crew	remarks
name	h UTC	h UTC	sec	sec	min	init	text

Ask for the data of your campaign if you need it.

A photo gallery is on the CD with the data.

1.2 Instrumentation

There are different reports from previous campaigns I can forward upon request. However, the key information was given in the above table of parameters.

2. Data processing

The raw data was processed using MetAir's software metair09.exe (Pascal under Delphi_2006/Windows-XP). The basic architecture can be explained easily: All raw data is treated in ONE step, with almost no metafiles (2 exceptions). The merging and transformation of raw to usable format (as delivered) is defined in ONE file called metair09.par prior to the execution of the program. A copy of metair09.par is mirrored automatically as DataVersionName.jou, whereas DataVersionName is usually the flight name with or without further pre- or postfix. These .jou are also part of the data set and documentation. Even if most content is quite specific and of no interest for the data users, there is some useful information at the end (after keyword "Comments"). And all the rest is at least for the sake of transparency (ask for details if needed). **No action to the data is done which is not visible in DataVersionName.jou!**

Here, we try to summarise the information which might be useful for the data user. However, if this is all too detailed, just proceed to the table with the description of the parameters, and ask us if you wish to have any modifications in the output, like different resolution, different parameters in different order, flags, etc.

2.1 Raw data

There were several sources (independent computers) generating 10 files (marked green in the list below), plus 2 metafiles during post-processing. All the "green files" were extracted from the binary versions of the data system already in Goulburn, and you got copies from them as well.

The following block can be found in any .jou (or metair06.par during execution):

E060521_1311 *(data set name)*
12 Files 50256 58102 0.05 ... *(number of raw data files, from-to time, resolution, etc.)*
E060521_1311.ANP 35986.0 0 *(raw data file name, zero time and clock drift [ppm])*
E060521_1311.ANV 35985.7 0
E060521_1311.ATA 35987.0 0
E060521_1311.ART 35986.0 0
E060521_1311.A1X 49819.3 25
E060521_1311.A2X 49819.3 25
E060521_1311.A3X 49814.0 25
E060521_1311.L75 49814.0 25

```
E060521_1311.LD9   49819.3   25
E060521_1311.FUG   0         0
E060521_1311.DTM   0         0
R060521_1357.RAD   -72.0     0
```

The blue numbers after the file names are defining the time offset either relative to GPS-time (where available), or additive to the zero of other time-tags. The offsets of roughly 36'000 seconds are to correct for Sydney time, the deviations from the 36'000 is 14 seconds for UTC versus GPS, and other minor adjustments (objectively adjusted during post-processing). The next column (red numbers) is correcting for clock drifts found during the sophisticated synchronisation process.

Table 1: Data sources and syncronisation:

ext.	from computer	from instrument	sync to/via	remarks
.ANP	center cube	Novatel (x,y,z)	same as .ANV	checked with z
.ANV	center cube	Novatel (vx,vy,vz)	.ART/vz	
.ATA	center cube	TANS Vector (att)	.ART/rol	
.ART	JREX	RT-3003	master	
.A1X	pod cube	REMlet in pod	same as .A2X	
.A2X	pod cube	BAT REMlet	.ART/accz	checked by wind calc.
.A3X	center cube	REMlet in fuselage	same as .L75	
.L75	center cube	LI-7500 RS-232	.A1X/uIRH2O	
.LD9	pod cube	Laser alt RS-232	same as .A2X	checked with zGND
.FUG	post process	FUGAWI	same as .ART	remark * below
.DTM	post process	SRTM	same as .ART	
.RAD	Rn sampler	Rn sampler	.A2X/press	
.RTX	internal storage	RT3003	master	format for May 22

After this header defining the input and it's synchronisation, the main instructions follow line by line (one line per internal output parameter). The line numbering is not necessarily sequentional. The parameter is exclusively defined by it's name, which can be given freely (case sensitive!). The units do not have any mathematical influence on the process, but, occur in the output. The three number columns are the identifier of the file, where the raw input can be found, then the column of this input, and finally the number of decimals in output (if ordered for output). After this, up to 10 instructions out of a collection of meanwhile 108 procedures (functions) form the final parameter. Some are self explaining. For others, even the author has to look into the source code sometimes. But, it is also a promise to ARA, that these functions will be shortly described in a future document.

```
  1   ptime [s]   1    1    2
  2   vtime [s]   2    1    2
  3   atime [s]   3    1    2
  4   lat1  [°]   4    1    5
  5   long1 [°]   4    2    5
...
...
222
```
(end mark for data processing)

This block (finished by "222") defining ALL parameters is used below to explain all the output, and how it was generated. However, here we proceed with explaining what can be found in .jou

The next block after the key-line "Output Files:" just defines which file formats are generated containing subsets of the above parameters in any order and time resolution. I do not explain all details here (e.g. the possibility to further filter output like picking a time sequence, and/or parameter range), but, just a few:

The naming "1hz.txt" means, that during the post-processing, this format is output as "1hz.txt" (trivial). But, the final version is then output as DataSetName.1hz (defined by the name in the very first line of .jou). "C" means that the time tags of the averages are interval centered, and 1, 60 or 0 defines the averaging times of 1 or 60 seconds, or maximum resolution respectively. The rest is straight forward picking parameters from the processing list. There are many other features like outputting flags (a coded history or "quality" of each data point), minima or maxima in intervals, or standard deviations in these:

Repeating a parameter (e.g. dw dw)causes the output of the parameter and it's Root Mean Square Deviation from the mean in the chosen interval. Putting a "<" as prefix delivers minima, and ">" maxima. The prefix "!" outputs the flags for this parameter. Since codes (e.g. waypoints, see parameter "frto" below) or flags are not meaningful as an average, one should always output them as minima or maxima.

An example of a file DataSetName.10s with 10-second intervals with time tags at the end of intervals could look like:

10s.txt E 10 zGPS w w <w >w <!w ///

and would contain zGPS, vertical wind w, its variance in the 10-sec-intervals, the minimum and maximum in these intervals, and the minimum quality flag (best point in interval).

Output Files:
1hz.txt C 1 within 1 TAS0 20 99 date clock edist long lat AUSx AUSy zGPS DTM zGND zLAS zInt MLt0 IRt0 temp0 pot virt IRpress press dewp e rH mix IRH2O IRCO2 kt15 sDN lDN sUP lUP >txVHF TAS0 GS1 vel dir TKE u v w avw avwL dw dwL avCO2 avCO2L avH2O avH2OL dCO2 dCO2L dH2O dH2OL CO2flux CO2fluxL H2Oflux H2OfluxL Mvel Mdir Merr slat slon salt spress power sflow stemp shum >spump >sflush >SAM rol1 tra1 tra2 nleg prad <frto FixPoints ///
xls.txt C 60 within 1 TAS0 20 99 date clock edist long lat AUSx AUSy zGPS DTM zGND zLAS zInt MLt0 IRt0 temp0 pot virt IRpress press dewp e rH mix IRH2O IRCO2 kt15 sDN lDN sUP lUP >txVHF TAS0 GS1 vel dir TKE u v w avw avwL dw dwL avCO2 avCO2L avH2O avH2OL dCO2 dCO2L dH2O dH2OL CO2flux CO2fluxL H2Oflux H2OfluxL Mvel Mdir Merr slat slon salt spress power sflow stemp shum >spump >sflush >SAM rol1 tra1 tra2 nleg prad <frto FixPoints ///
20h.txt C 0 edist long lat AUSx AUSy zGPS DTM zGND zLAS zInt temp0 press mix IRH2O IRCO2 vel dir u v w dw dCO2 dH2O rol1 nleg <frto ///
end of list *(end mark, all following output formats are ignored)*

There is an option to add FixPoints with coordinates, and the radial within which the closest passing is searched (see parameters frto, prad and FixPoints at the end of the main parameter table):

FixPoints:
IsoTrans.fix *(file name where the landmarks can be found)*

Plots: *(all available plot templates, where those without a "/" are executed after running metair06.exe)*
samplemap.grf
/windmap2.grf
/tdwprof.grf
...
/nomoreplots

Comments *(after this key line, anything descriptive can be added, see remark *** below)*
IsoTRANS-06/8
Jorg/Bruno, with photos.
standard pattern with 6 Rn samples.

notes from flight-log:

notes from post-processing:
060613 basic adaptations to MetAir post-processing finished, including three different wind calculations.
060615 trends in IRH2O removed.

***) This option was not yet used intensively, but, I strongly recommend to use this option in order to keep the .jou as a concise, complete, and actual document, in which the history and the findings to each flight are conserved.

After this rush through the general .jou-format (or metair06.par) the details to each parameter are explained in the following comprehensive table using an excerpt of E060521_1311.jou as it's seed (this flight was processed first).

Appendix 1 (not yet checked $$$)

A few words on flagging data points (etc.)

This text started as a small internal note when trying to introduce quality flags in the data set. Those, who only wish to understand our new flagging of data just go to "Back to focus", or even to "Caption for data sets with MetAir data quality flags" at the very end. The rest are introductory remarks, which are regarded as a contribution to the discussion between us, and the data users.

The "decision tree" of the question "how to introduce flags" grew very quickly into a jungle, and on the other hand, some "old wisdoms" were confirmed. Before the full story which evolved, I wish to give a summary of my (subjective) thoughts:

1. Flagging data is a necessity, but, it's impossible to do without focusing, and without compromises (in a realistic framework).
2. Any data set (in some cases any flight) is different in it's scientific aim, and hence the prime parameters (errors in a specific parameter might be of no importance for one project, but, catastrophic for another).
3. There is no ideal flight where all data is perfect; we (and the post-processing program) have to cope with more or less imperfection.
4. A data set is never "finished"; data quality is an iterative process, with an asymptotic end.
5. The best way to cope with all these deficiencies is a continuous communication between aircraft/instrument operator, and data users. A "backbone" of this communication is writing/reading/updating documentation.
6. Trying to normalise or regulate this complex interaction ("good practice is ONLY, when ...") is not very helpful. Time should not be wasted for regulation, and efforts to comply to, but, should be used for improving the individual data sets, and the documentation.
7. When errors (not only measuring errors) are identified, conclusions for further projects (and their budgets!) should be made.

The detailed discussion, focused on the questions of quality flags, beginning with an excursion

First, there is the wish to have a maximum of quality information accompanying any data point. Everybody would agree, that it's "normal" to have error bars, flags, when there is a problem, etc. That's what every student is learning. It's not bad at all to repeat these "basic things" sometimes, and to try to comply to. But, on the other hand, field work as a part of real life is more complex than an average student is assuming, and therefore, the practice, measured relative to these high goals, might sometimes be disappointing. Sometimes, even "trivial" things, like "what is an altitude?" can cause a very long, absolutely non-trivial discussions, ending with facts like the shape of the earth, and the inhomogeneous gravity field.

So, realising such high goals in "quality declaration" is very ambitious. Let's begin with the easy things. First, there is the commonly known distinction between accuracy and precision (absolute, calibrated accuracy versus short term reproducibility). These can be handled - already with shortcuts, assumptions, and compromises - by filling tables, characterising the parameters measured. Already these "well defined, easy to declare" quality characteristics are not trivial when changing from theory to practice: Both are extremely depending on pre- and post-flight calibrations, maintenance - to mention only two which can be influenced by the operator (and the budget) -, but very often are strongly depending on flight conditions, hence, individual flights. Some parameters might be affected by fast descents or ascents. Others, like the wind vectors, are sensitive to curves (not in theory, but, in practice ...).

Then, additional criteria might be important, as e.g. the length of undisturbed tracks (no gaps in the data, no inhomogeneities in the atmosphere), or the spectral behaviour of the data.

To be strict in all these questions, the operation of a research aircraft would either require an order of magnitude more personnel than available, and/or much more than 24 hours during a day in the field, plus several days or weeks for post-processing. Or, one has to flag any data point as "suspicious". The dilemma can only be solved by a pragmatic approach and a lot of compromises, based on experience, and assumptions of what is regarded as important. For the latter, the communication with the user is most important. Unfortunately, this discussion - if ever - only starts AFTER a campaign. In an IDEAL WORLD, this would start earlier (instead of dealing with other project administration ...).

In some cases, the focus is easy to define, e.g. in Aerocarb/CarboEurope, it's clear, that the prime concern is the accuracy of the CO2 profiles (both from flasks, and from in-situ). Almost all the rest is "nice to have". In other projects, it's not so clear from the beginning, what is of prime concern. This is even more true in cases, when data sets of one project could perfectly be used by another project, if only the data quality of a parameter regarded as "unimportant" would have been calibrated. So, the general goal is to maximise the data quality of all parameters measured. In campaigns, where we tried to learn something about photochemistry, the accuracy of many parameters is important. But, I almost do not dare to write something like this, because - again - it's almost impossible to reach. Planning, performing and processing multi-parameter research flights is a permanent fight against the reality, in order to reach the most important portions of what was desirable, if possible on time (some pilots even say, that flying is a permanent fight against falling down - so our work is fighting against all odds all the time).

So, dear customers and students reading this and using MetAir data, be patient with "final, well documented data". A good compromise is reachable, and what we can reach today (May 2004), was not possible before. But, we always need your help, and time, and sometimes even good luck.

Back to focus:

When I started with the first lines of program code of "metair04.dpr" (Delphi Project) about 5 months ago, I introduced an array of byte, covering the whole parameter space, reserved for flagging. Now, where the program does about everything I have planned and many flights are already processed at least preliminary, I wish to introduce "flagging". But, that is not trivial at all. For a consequent flagging, which is, that every number in the data file receives a code, from which the processing can be backtraced, and the quality can be estimated, I would need more variables (storing space) than for all the other 150 to 200 parameters, with a resolution of 10 Hz during up to 5 hours. And in the output, the data would be buried under flags. This is, because, in principle, any of these variables can be treated (affected) by up to about 80 procedures, which might leave a spur in the data quality. When using a certain parameter as input for calculating another (e.g. for wind more than a dozen), the number of flags and their meaning would multiply. Therefore, we need to reduce. I decided to try the following approach:

First, from all the available procedures, I identify only those (12), which really could harm a result in a way, the user would like to know. All the other procedures from trivial calibration polynomials up to complex wind calculations are assumed to be "standard". Additionally, these parameters are grouped in order to have not more than 8 kinds of flags (for the one byte code). And, the code should be both precise and intuitive, i.e. when the user is not ready to deal with the details, he/she should know, what is the qualitative meaning. I have chosen: The higher the code, the lower the data quality (or the higher the level of suspicion).

Suspicion is another keyword, I have to introduce an excursion for:

We all wish to have a result of a measurement, and a quantitative measure of it's precision/accuracy. This is possible, e.g. as "worst case error bars", or as RMSD when there is any redundancy (like flasks with laboratory analysis against in-situ). But, it's not feasible to apply to all data, unless one declares very large error bars (temperature +/- 2°C, CO_2 +/- 2 ppm, NOx +/- 50% of the reading, etc.), making the data set useless. This is not necessary, because in most cases, the quality is much better (typically about a fifth of the above examples), and the exceptions are rare (even more rare, but, not to exclude: larger errors). So, we have to live with a semi-quantitative expression of "level of suspicion". A good example are the fully quantitative error estimates for the three wind components. It's an individual, point to point information in m/s, pretending to be an error bar. But, it's not! It's just a number, expressing the residual between the acceleration measurements, and the GPS 3-D velocities. Low values certainly are suggesting reliability of the results, but, it's no guarantee on the one hand, and wind wind vectors with very large "errors" can be perfect (when the acceleration and attitude was perfect, but, the GPS information was incorrect). It's just a practical compromise: High values are also indicating curves, or other transient flight phases which might have to be excluded for some specific results (again: This is strongly depending on what should be done with the data).

In earlier attempts to declare (and filter!) data quality, I ended up with data sets with more holes in it than data. Now, I have changed this general attitude: I try to estimate a parameter, even when it's lost, if there is a way to do so. A good example is the roll angle as a prerequisite for any wind calculation (the angle the wings have relative to the horizon): Without the direct attitude measurement, it's possible to estimate a +/- 2° roll angle by just differentiating the rate of change of heading. I tested this during several flights and manoeuvres. It works astonishingly well, but, is certainly depending on the assumption, that the shift angle is near zero, or at least constant. With the introduction of such features, the need for flags grew, because one wishes to know, whether a certain number in "rol" is measured, or just estimated. When I am claiming, that our new system is "robust", I am always remembering a former Professor stating that "robustness is a synonym for insensitivity (and stupidity)". With this compromise of filling gaps with estimates, and flagging them, I hope to offer the possibility to any user to choose his level of sensitivity.

Also by doing this, another compromise had to be done: In principle, one could only flag instead of throwing out data by filtering (e.g. the de-spiking in certain parameters). But, this is not practical, because the user wishes to plot and/or use "nice data". But, with the new architecture of the post-processing, it's absolutely possible to get unfiltered data with flags, or both (filtered and unfiltered). It's – again – just a matter of communication between the user and us, and of re-processing the flight (usually a matter of 15 minutes when everything else is in "final stage" - for a flight not treated yet this is still approximately one day work).

Caption for data sets with MetAir data quality flags:

Considered the value space of 0..255 for the flagging byte, we have the decimal numbers 1, 2, 4, 8, 16, 32, 64, and 128 to express the different actions. To reach the goal of having an intuitive flagging (the higher the worse), we have to introduce a hierarchy into the procedures/groups. We have to use flag 1 for the faintest and 128 for the worst potential deterioration of data quality. This is - repeated once more - depending on the campaign, but, I have chosen the following as standard (which can be changed for specific data sets):

1: Indirect "spur of former flags" in secondary parameters, when e.g. a temperature in an integrated pressure altitude had a flag of >4 (Flags of important input parameters like the pressure for a pressure altitude are fully transparent, i.e. when the pressure had a flag "2", then also the altitude gets it.

2: Averageing and smoothing of data. This is usually "harmless" and fully intended, but, might be a problem for certain applicatins. At least it is "nice to know". So, the user knows, that the smoothing could be changed if necessary.

4: Interpolation of data. In most cases, this is absolutely "harmless", because e.g. data from an instrument outputting 1-Hz-data needs interpolation when integrated in a 10-Hz data set. It's the responsibility of the user to judge, whether an interpolation is too long for her/his application. In the procedures, the maximum gap length to interpolate can be specified. So, any adaptation is possible, but, has to be discussed.

8: Filling of gaps with estimates (like the above example of roll angle, or gaps in the temperature measurement filled by some backup sensor), or values off the different NOxTOy channels modified using the equation $NO_2 < NO_x < NO_x+PAN < NO_y-HNO_3 < NO_y$, or $O_x = O_3+NO_2$;

16: Values introduced manually.

32: Reference to an important note in the data description.

64: Value filtered out by some automatic filter. The values do not show up in the data set (default -99), unless ordered specifically.

128: Value eliminated manually (the value is set to -99).

Because we have chosen individual bits of an 8-bit byte which are expressed as decimal numbers in the data sets, the flags can be added without loosing their identity. It's not so easy to break them apart during reading "by eye", but, it's trivial to identify them in a program, which is further treating the data.

For quick use we can take as a rule of thumb: All data with flag < 8 is ok. With values between 8 and 63, one has to be careful for certain applications. Extending the rule of thumb to the wind "error estimates", I recommend a limit of about 2, where below, the data is ok (with a precision of 0.5 to 1 m/s, not only 2 m/s). In order to make it "easier" for the user, the values of the flags could be normalised (divided) in a way, that the limit is 1.0 (ok below, suspicious above). But, I wait before introducing too much sophistication. But, the structure, once implemented, would certainly allow such a modification.

We only insert those flags in the data sets which we regard as essential for the specific user. But, based on the full documentation, containing also much more than the parameters selected for output, the user can choose and order additional flags.

At the end, let's discuss another tricky practical detail: Flags are well defined above for individual data points in the fully resolved (5 or 10 Hz) data set. But, how do we handle averaging? The standard is now the minimum flag within the interval. So, a 1-second-average of 10 individual measuring points, where 3 were thrown out by automatic filtering, there is no flag for filtering anymore. Some might wish to output the worse instead of the best flag for an interval. Also this can be chosen individually.

I am sure to encounter more practical problems when working with all the data which is still in the tube, but, I hope that we always find pragmatic ways to meet the needs of the data users.

6 May 2004, Bruno Neininger, bruno.neininger@metair.ch (+41 41 755 38 04).

Appendix 2: Modified excerpt of a former report (Interreg 2003) concerning different levels of "results"

We already discussed the principal advantages and some limitations of airborne measurements. In the following, we outline the way from raw data to final results. This way is usually longer than one would expect. And when the final data is ready, one has another dilemma: One can not inspect all the data with all it's details. It's too much, and 90% is "boring". So, one tries to find the most interesting features detected (case studies), and to derive some general results from overviews like scatter plots or tables.

Here, we discuss some aspects of the data collected, well knowing that we have to do shortcuts which might raise questions. Some questions are possible to answer by the reader when he/she is looking into the appendices, where the data is displayed, or by looking into the data by the means of own graphical data analysis programs or models. Therefore, we have put great emphasis in presenting the data in very comfortable formats, and tried to describe the details in the documentation which goes with every archive.

The whole development of results can be looked at as a pyramid. On top, there is the key result, but, there are many layers below, that are more or less developed in this report or in the appendices or in the data set. This pyramid has the following layers:

```
L9:   /\                THE key result(s);
L8:  /__\               a vast variety of different direct results;
L7:  _____\             organised display of data;
L6:  ______\            selected parameters in homogeneous formats;
L5:  _______\           additional derived parameters (e.g. fluxes);
L4:  ________\          about 140 primary parameters plus VOC's;
L3:  _________\         individual, calibrated datasets;
L2:  __________\        individual, non calibrated datasets;
L1:  ___________\       synchronised, filtered raw data;
L0:             \       raw data from 5 on-board computers (8 files);
```

The way from level L0 to the top is not a direct one. There is much iteration involved. Sometimes, one realises on level 6 or 7, that it might be useful to modify the fundament L1. On the other hand, an idea about what to expect on level 8 sometimes is developing already during a flight, when we observed on the real-time data display in the cockpit e.g. that we crossed a pollution plume somewhere, or that the stratification is special, and many other details. We try to note such findings during the flight either in the electronic logbook, or as a hand written note (both are included in *.jou !)

However, since 2003, the steps from L0 to L6 are all "frozen" in the *.jou: Travelling back and forth between levels 0 and 6, and for prepared graphics even to level 7 is fully automated and quick to modify. It's all controlled by a single, easy to read ASCII file, where all the instructions are stored, and even the logbook entries and other notes from before or after the flight, including remarks from the data processing, are written to.

The architecture of this pyramid does not know any metafiles, where pre-processed data is stored, or modified, and where one would have to take care which versions are in use. Once the parameter file is tailored, any level until L6 is generated automatically by a single run of one program.

Also the ascent from L0 to L1 ("correction of raw data") is nothing mystical, but, includes actions like synchronising different data sources, filling gaps with interpolations or redundant parameters, adjusted filtering, smoothing, etc. Because data sets are usually heterogeneous, the flexibility of this first step was important. The basic data set never changes, and there is absolutely no manual editing of data involved. Even more: Any procedure (e.g. filtering) applied to any data point is expressed with flags collected in the file *.flg (see appendix about flagging). So, the data set is fully transparent. However, to use all these features is quite time consuming. Most data users might not take the pain. But, it's good to know that any data point could be traced back to the raw data.

A complex part of step L1 to L2 is the wind calculation, where a variety of primary parameters such as the attitude angles (the three angles of the aircraft relative to the earth), accelerations and flow measurements are merged.

The ascent from L2 to L3 was done, when all the pre- and post-flight calibrations were processed.

L8 is usually being generated when preparing L7, when looking into all the graphs. But, this is still so much on many time scales, that it's impossible to note them all and to discuss them in this report.

The "final climb" to the top level L9 is usually done when writing joint publications. Because all the levels below are a non-trivial first step I expect to be asked as a co-author of any publications discussing these results.