

The Douglas Argos-Filter Algorithm

PROLOGUE:

This SAS version (8.50) of the Douglas Argos-Filter (DAF) is qualitatively equivalent to the algorithm that is now freely available through Movebank (www.movebank.org), and described in a 2012 article published in *Methods in Ecology and Evolution*:

Douglas, D.C., R. Weinzierl, S. Davidson, R. Kays, M. Wikelski, and G. Bohrer. 2012. Moderating Argos location errors in animal tracking data. *Methods in Ecology and Evolution*. doi: 10.1111/j.2041-210X.2012.00245.x

(journal paper, also bundled in the version 8.50 distribution package)

<http://onlinelibrary.wiley.com/doi/10.1111/j.2041-210X.2012.00245.x/abstract>

(Movebank filtering video tutorial, also available at the www.Movebank.org website)

http://www.youtube.com/watch?feature=player_embedded&v=iJha6OzM4UY

This version (8.50) of DAF differs from the prior distribution (version 7.03) in two ways: 1) great-circle distances are calculated using the WGS84 spheroid, so distance calculations may differ by a fraction of a percent; and 2) the method of choosing between the Argos primary and alternate location solutions is greatly improved, meaning less need to use the 'hardcode_changes.txt' method (see the MEE paper above, and kudos to Rolf Weinzierl). Differences between the filtered outputs of Version 7.03 and 8.50 would likely be very few and/or qualitatively inconsequential, so don't be overly concerned if you've used and published your studies using prior version(s).

Version 8.50 also includes SAS code and public-domain software (ttermpro) that can be configured to conduct Telnet sessions that download raw DIAG and DS data (i.e., live-update) (See Appendix A). It is then convenient to configure the Windows Scheduler to run the DAF on a fixed schedule (for example, daily or weekly) during a period when animals are being actively tracked (see Appendix B). This keeps a data base up-to-date.

Version 8.50 also has SAS code and public domain software (blat) that can be configured to email the filtered results to a user-defined recipient list. For the email to work, the user must have access to an SMTP server that does not require a secure shell interface. Many SMTP servers, however, require secure interfaces (for example, GMAIL). For such servers, the user can install a public domain software called "stunnel", and then for example, use GMAIL's SMTP capability to route emails. More details about setting up the Windows Scheduler, and setting up email can be found in the Appendices B & C, respectively.

It is worth noting that both live-update and emailing are features that can be set-up in Movebank. Essentially, Movebank's DAF capabilities were developed to emulate the SAS implementation of DAF; yet Movebank is free and it offers many other advantages for data sharing, data archiving, and data analysis.

ABSTRACT:

The Douglas Argos-Filter (DAF) Algorithm ingests Argos¹ satellite tracking data (*DIAG format*² only) and flags improbable locations based on user-defined distance and velocity thresholds. The algorithm writes all results to ASCII comma-delimited files for spreadsheet or GIS applications, as well as *KML 2.0*³ files compatible with *Google™ Earth*⁴, but XML files compatible with *NASA World Wind*⁵ have been discontinued. The entire unfiltered data set is also output, so in its minimum application, the algorithm is a useful tool for parsing Argos *DIAG* data into a more convenient digital format.

Service Argos disseminates each location with a location class (LC) accuracy index that falls under two broad categories: *standard*⁶ and *auxiliary*⁷. Standard locations (LC 3, 2, 1) have an *estimated 1-sigma error radius*⁸ of 250, 500, and 1500 m, respectively, while accuracy of the auxiliary locations (LC 0, A, B, Z) is highly variable and undocumented by Service Argos. Auxiliary locations are largely unsuitable for interpreting animal movements without first addressing accuracy issues.

The DAF assesses plausibility of every Argos location using two different methodologies based on: 1) distances between consecutive locations; and 2) rates and bearings among consecutive movement vectors. Both filters independently move chronologically through the raw tracking data of each individual animal, evaluating 3 locations at a time. Results of both filters are written to separate output files. The first filter searches for spatiotemporal redundancy, under the premise that significant location errors rarely occur in the same geographic locale consecutively. The user inputs a distance threshold that defines the spatial scale of "locale". The second filter constructs 2 adjoining vectors and evaluates their velocity, internal angle of intersection, and length, under the premise that significant location errors are commonly associated with tracking signatures that create implausible movement rates or anomalously acute movement angles. User-defined thresholds for implausible velocity and suspect angles allow the user to adjust the filtering to better conform to a species' movement capability and behavior. Depending on species, one of the two filtering methods typically possesses advantages over the other. For avian species, the algorithm also produces a third output that optimally merges the two filters, since birds intermittently migrate long distances with high velocity and directionality, but during non-migratory periods they tend to occupy relatively localized nesting, molting, staging, or wintering areas.

The Douglas Argos-Filter Algorithm is available from the *USGS Alaska Science Center*⁹, however, the program requires a *PC-SAS*¹⁰ software license for execution.

¹<http://www.cls.fr>

²http://www.cls.fr/manuel/html/chap4/chap4_5_2_4.htm

³http://earth.google.com/kml/kml_intro.html

⁴<http://earth.google.com>

⁵http://www.worldwindcentral.com/wiki/Main_Page

⁶http://www.cls.fr/manuel/html/chap2/chap2_3.htm#2.3.5

⁷http://www.cls.fr/manuel/html/chap2/chap2_3.htm#2.3.6

⁸http://www.cls.fr/html/argos/general/faq_en.html

⁹<http://alaska.usgs.gov>

¹⁰<http://www.sas.com>

PURPOSE :

Ingests any number of Argos DIAG files and reformats all information into ASCII comma-delimited files, the selects the most plausible location between the ARGOS primary and alternate locations based on minimizing total track distance. Flags implausible locations based on two different filtering methods:

- 1) Minimum Redundant Distance (**MRD**): a user-defined distance threshold for determining location redundancy;
- 2) Distance, Angle and Rate (**DAR**): measurements that attempt to identify implausible locations based on the fact that most suspicious ARGOS locations cause an animal to incorrectly move a substantial distance and then return, resulting in a tracking-path that goes 'out-and-back' (and/or further validated by unrealistic movement rates, depending on the temporal frequency of the locations);

Hybridization (**HYB**) of the **MRD** and **DAR** filtered results is also created. The hybrid was specifically developed for avian tracking that contains relatively high-speed, directional, migratory events. Locations that passed the **MRD** filter are retained as 'anchor points', then chronologically intervening **DAR** locations are evaluated to determine if they adhere to directional movement when compared to the vector formed by their preceding and subsequent **MRD** anchor locations.

REQUIREMENTS :

A) PC BASE SAS - Version 7 or later.

This SAS program requires PC SAS because it uses DOS commands to concatenate all your individual ARGOS DIAG files into a single file before ingestion. This concatenated file will be written into the same directory where your SAS program files reside, and will be named 'total.dia'. It can be considered a temporary file, which you may delete after the SAS program executes.

B) All your Argos ASCII-format DIAG data files in ONE directory. The SAS program has been written to accept DIAG files from CD-ROM, Telnet, and/or ADS (e-mail), in both compressed and uncompressed formats. A SAS program that executes Telnet sessions with CLS is included in the program distribution package. This program is convenient for automating the acquisition of raw DIAG and DS files.

C) You must create an ASCII text-file that defines each of your ptt deployments. This file is comprised of IF-THEN-DO logic blocks relating each ptt-number to the respective animal upon which the ptt was deployed, and during what time interval.

D) You must modify the SAS program '00_template_runv850.sas' to supply several user-defined thresholds that establish minimum distances, maximum rate of movement, etc., which largely dictate how conservative or liberal the filtering will be implemented. You must also define several directory pathnames (specific to your computer) that define where your input data are stored, the name and location of your deployment definition file, and the directories where results will be written.

CAVEATS :

If this program filters a location, it does not necessarily mean that the location was incorrect. Conversely, and more importantly, if this program does not filter a location, it does not necessarily mean the animal visited that precise location.

The underlying premise for the **MRD** filtering logic is: 'Argos does not make a significant error in the same place twice, consecutively.' In reality, bad Argos locations sometimes do occur at the same locale, consecutively, but with a low probability of occurrence. In about 2000 control relocations, I witnessed 3 consecutive pairs of locations that would have passed typical **MRD** threshold criteria (but the Argos Multi-Satellite Service appears to have increased this probability). So, if you encounter suspicious results from this program - then question those results.

This program will quickly (and systematically) get about 95+% of your Argos data "cleaned-up" to your satisfaction, but you should definitely invest some QC time reviewing the results and getting any last anomalies taken care of before you make and publish biological interpretations.

You control how conservatively the filters behave through user-defined thresholds. How conservative (or liberal) the program is allowed to behave should be commensurate with the study's spatial scale-of-interpretation.

This program does not correct the inherent location error of any single Argos location. The program strives to reduce the average error among all locations that pass the filtering criteria.

GETTING STARTED:

1. Download the Version 8.50 distribution package (ZIP file). Save the ZIP file at the top-level of the disk drive that you will use to store and process Argos tracking data. For example, the root level of your C:\ drive, or D:\, or F:\, or whatever - the important issue is that you must have write access and ample available disk space. Below, I will refer to the drive as "X:\".
2. Unzip the distribution ZIP file into the drive's root level. This will create a sub-directory hierarchy X:\argos\projects\template
3. Under the .\template directory, there will be a suite of other subdirectories. I strongly recommend you do not change anything under the .\template directory, rather, use its contents as a template.
4. Create a sub-directory under X:\argos\projects that is more meaningful to your study, for example: X:\argos\projects\swans_2012
5. Copy all of the folders (and their contents) that are in X:\argos\projects\template into X:\argos\projects\swans_2012
6. Now, customize the files under X:\argos\projects\swans_2012 as described below. Everywhere you see "template", substitute "swans_2012". This includes file names as well as global substitutions in the 00_template_runv850.sas (which you would have renamed 00_swans_2012_runv850.sas"). Repeat steps 4-6 for other existing projects, or future projects.
7. **DO NOT**, however, make any changes what-so-ever to the files (and file names) that are under the subdirectory named: .\core_sas_argos
8. Modify the google_kmz_credits.txt file in the .\sas directory. This information gets embedded into the Google Earth KMZ file so it contains a form of acknowledgement to project personnel.

IMPORTANT → READ THIS**User-Manual Color-coded Syntax:**

SAS Program code that you really should consider changing is highlighted in a **bold-red font** below.

SAS Program code that you might consider changing is highlighted in a **bold-blue font**.

SAS Program code that could safely remain default is highlighted in a **bold-magenta font**.

SAS Program code that is purely optional is highlighted in a **bold-green font**.

Do not change anything that isn't color-highlighted, unless you understand SAS syntax.

USER DEFINED PTT DEPLOYMENTS:

You must create an ACII text file that describes each of your PTT deployments. This file is written in SAS code, in the form of IF-THEN-DO logic blocks - one for each PTT that was deployed on an individual animal.

Probably 95% of the emails I receive from users that are having difficulty getting this SAS program to execute are related to syntax errors they introduced into their deployment-logic-block file. SO, READ THESE INSTRUCTIONS CAREFULLY, and then verify the syntax of each and every line of your file.

THE PTT DEPLOYMENT LOGIC BLOCK FILE:

A file named "template_deployments.txt" is distributed with this program, and provides 2 logic-block examples that can be used as templates to copy/paste.

One block of if-then-do logic is required for each ptt deployment; copy-paste as necessary. A unique ANIMAL code (25 characters maximum, simple ASCII (A-Z, 0-9, or the underscore character, NO blank spaces, NO apostrophes, NO special characters) must be assigned - make something up if you don't have meaningful animal IDs. ANIMAL is a character variable, so it must be bounded by single quotes. The first datetime value is the date and time when the ptt was deployed on the animal. NOTE: the time must be entered in GMT (UTC/ZULU) if the **GMTOFFSET** value is 0 (zero). If you have chosen to offset all date/time values to correspond to a 'LOCAL' timezone, then the deployment time must also conform to that LOCAL timezone. The deployment datetime is entered twice (2nd entry defines the variable DTIME). ALL Argos data prior to the deployment datetime and after the end-deployment datetime will be REMOVED from ALL output datasets.

DO NOT CHANGE the variable names, only change their values. The declared 'end time' (2nd date-time entry) should be set well into the future. All data after this time will be omitted from the output files. You will need to insert a real 'end time' if you have deployed a given ptt number on more than one animal over the course of your database/study. You may also want to insert a real 'end time' if your animal died, and you want to remove all subsequent 'dead animal' data from the output data sets.

You must specify the deployment location. The latitude (deplat) and longitude (deplon) must be in decimal degrees (west longitude and south latitude must be negative numbers).

You may optionally include any other animal-deployment specific variables, such as sex, pop, age, etc. They will be carried-through to the output files. You can name the optional variables as you like (8 characters or less), and you should include a single format statement (for numeric data) or length statement (for character data) before the deployment logic blocks, as illustrated below.

NOTE: MONTH is coded with 3 characters following SAS convention: JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC. Do not deviate from this convention.

This file must be saved in ASCII TEXT format. Create and edit the file using a pure ASCII editor such as NOTEPAD, or if you use word processing software (e.g. MS-WORD), then be sure to "save as" ASCII text. Note the full-directory pathname and filename of this file; you must enter that information into the SAS program as described below.

See the accompanying "template_deployments.txt" file for functional examples and more instructions.

- * Note: If you add your own variables, declare the variable-format(s) first.;
- * Note: Adding more variables is optional; and all variables are retained in the output files.;

```
format weight 4.1;
length sex $ 1;
if ptt = 16976
  and thetime gt '12NOV1998:11:30:00'dt
  and thetime lt '31DEC2002:00:00:00'dt then do;
  animal = 'FX1998 16976';
  dtime = '12NOV1998:11:30:00'dt;
  deplat = 70.255;
  deplon = -148.368;
  spp = 'Common name or species';
  colorname = '25 choices, see readme directory';
  icontype = 'circle';
  sex = 'M';
  weight = 9.2;
end;

```

<- PTT number
 <- Datetime of ptt deployment
 <- END of deployment datetime
 <- Unique animal ID (character)
 <- SAME datetime as deployment
 <- Latitude of deployment +N/-S
 <- Longitude of deployment +E/-W
 <- Common name or species name
 <- GE color, see win_colors25.pdf
 <- circle or square, GE icon shape
 <- Optional parameter(s) that will
 appear in your output files
 <- End of logic block

USER-DEFINED FILTERING PARAMETERS:

Details about each user-defined defined parameter are presented below. These parameters control the filtering results - a few are very important. You should read the documentation below for each. These metrics are also described in the Methods in Ecology and Evolution paper, as well as the paper's supplemental materials -- both are in the `.\readme_files` subdirectory.

```
%let minoffh = 12 ;
%let maxredun = 10 ;
%let minrate = 110 ;
%let ratecoef = 25 ;

%let gmtoffst = 0. ;
%let latmin = -90. ;
%let latmax = 90. ;
%let lonmin = -180. ;
%let lonmax = 180. ;

%let r_only = 0 ;
%let r_or_a = 1 ;

%let keep_lc = 1 ;

%let rankmeth = 2 ;

%let keeplast = 0 ;
%let pickday = 0 ;

%let skiploc = 0 ;

%let xmigrate = 2 ;
%let xoverrun = 1.5 ;
%let xdirect = 20 ;
%let xangle = 150 ;
%let xpercent = 20 ;

%let testp_0a = 2 ;
%let testp_bz = 3 ;
```

COMPUTER-SPECIFIC MANDATORY CHANGES:

You must adapt the SAS code to conform to your computer system setup, most notably the directory pathnames.

```
* SET VARIABLE offset TO THE NUMBER OF HOURS YOU NEED TO ADD TO GMT TO ACHIEVE YOUR PC
CLOCK TIME;
%let offset = -9;
```

```
*****;
* USER ASSIGNED DIRECTORY PATHNAMES: ;
*****;

*****;
* Specify the directory name you assigned to this project, ;
* first bounded with single quotes, then again without the quotations. ;
* Do the same for the disk-drive-letter that the project resides on. ;
* It is assumed you have established on your hard drive a directory ;
* path that follows this convention: ;
* ;
* In this distribution SAS program, the project name is template ;
* ;
* <DRIVE>:\argos\projects\<<project-name>\ ;

%let driveletter = 'd:' ; /* <DRIVE> in quotes */
%let dletter = d: ; /* <DRIVE> without quotes */

%let projname = 'template'; /* <project-name> in quotes */
%let pname = template ; /* <project-name> without quotes */

*****;
*Directory containing ONLY ASCII-format Argos DIAG format data. ;
*First, the <drive-letter> in single quotes. ;
*Second, just the directory pathname without the drive letter, quoted. ;
*Third, the full pathname (combine the 1st and 2nd), without quotes. ;
*****;
%let diadrive = 'd: ';
%let diapath = '\argos\projects\template\rawdiag';
%let diafull = d:\argos\projects\template\rawdiag;

*****;
*Directory containing ONLY ASCII-format Argos DISPOSE format data. ;
*First, the <drive-letter> in single quotes. ;
*Second, just the directory pathname without the drive letter, quoted. ;
*Third, the full pathname (combine the 1st and 2nd), without quotes. ;
*If you do not have DISPOSE data, that is OK - just duplicate the same ;
*disk drive and paths that you used above for the DIAG data. ;
*****;
%let disdrive = 'd: ';
%let dispath = '\argos\projects\template\rawdisp';
%let disfull = d:\argos\projects\template\rawdisp;
```

```
*****;
*FULL pathname (with the filename) of your deployment logic block file.;
*NOTE NOTE NOTE: This file MUST exist, and YOU must create it.      ;
*SEE the accompanying file template_deployments.txt for doc.        ;
*****;
%let deploys = d:\argos\projects\template\sas\template_deployments.txt;

*****;
*A synopsis of the most recent overpasses and most recent locations  ;
*is created. Nrecords, below, establishes the number of overpasses  ;
*and locations that will be reported for each ptt.                  ;
*****;
%let nrecords = 10;

*****;
* SAS CODE TO BUILD MOVEBANK-READY INPUT FILES                       ;
* Put an asterisk in front of the %include statement to skip this. ;
* Preparing file for uploading into Movebank (www.movebank.org)    ;
* is not mandatory. It is best considered at the end of a tracking  ;
* study after all data have been collected. It is also convenient  ;
* making consolidated DIAG and DS data files in which overpass    ;
* redundancy has been removed.                                     ;
*****;
%macro make_movebank;
%include mb_arg;
%mend;
*****;

*****;
* SAS CODE TO EXECUTE A TELNET SESSION AND DOWNLOAD NEW ARGOS DATA. ;
* Put an asterisk in front of the %include statement to skip this. ;
* If realtime acquisition of Argos data is desired (a live update) ;
* then modify the get_argosdata_telnet.sas program accordingly and  ;
* remove the asterisk in front of the %include statement below.    ;
* Directions for modifying the get_argosdata_telnet.sas program are ;
* embedded as comments directly within the SAS program itself.    ;
*****;
filename dotelnet "&dletter.\argos\projects\&pname.\sas\get_argosdata_telnet.sas";
*%include dotelnet;

*****;
```

CHANGES THAT ONLY AFFECT THE GOOGLE EARTH KMZ FILE:

```

*****;
* ALL THE REMAINING USER OPTIONS CONTROL ASPECTS OF THE GOOGLE EARTH ;
* FILE THAT CONTAINS THE LOCATIONS THAT PASSED THE FILTERING CRITERIA. ;
*****;

*****;
* Define the filtered locs that will appear in the POOLED GE KMZ files ;
* Set the 2 %let statements below to any pair of options below. ;
* ;
* FOR THE DAR filter method, use... ;
* outdar - locations that passed the distance-angle-rate filter ;
* pikdar - subset of above, but only the best location per duty cycle ;
* ;
* FOR THE MRD filter method, use... ;
* outdis - locations that passed the redundant distance filter ;
* outpik - subset of above, but only the best location per duty cycle ;
* ;
* FOR THE HYBRID filter method, use... ;
* outboth - merged hybrid of the outdar and outdis datasets ;
* pikboth - subset of above, but only the best location per duty cycle ;
*****;

%let kmltype = outboth;
%let kmlpick = pikboth;

*****;
* NOW select the corresponding prefix for the GE point annotations ;
* use DAR or MRD or HYB depending on what filter you chose above ;
*****;

%let filtpref = HYB;

*****;
* For Google Earth and the World Wind "daily date" layer, ;
* if you want to weed occurrences of the text-date annotations to only ;
* the endpoint locations of vectors that exceed a length threshold, ;
* OR for vectors whose time interval exceeded duration threshold, then ;
* set the weedlable variable to ONE (1), and define the vector length ;
* threshold (km), and the duration threshold (hours). ;
*****;

%let weedlabel = 1;
%let daydate_dist = 25;
%let daydate_hours = 24;

*****;
* For Google Earth, set desired scale of the date-text annotation ;
* that is written next to each location, and for the LAST location. ;
*****;

```

```
%let txtscale = 0.60;
%let txtscalelast = 1.5;

*****;
* For Google Earth, set desired scale of the icons (symbols) that plot ;
* each location. Then, for the last location, scale up by what factor. ;
* Next, set the line width (units=screen-pixels) of the vectors that ;
* vectors that connect the locations. ;
*****;

%let ge_iconscale = '0.4';
%let ge_lastscale = '1.5';
%let ge_linewidth = '2';

*****;
* For Google Earth, set desired opacity of the line vectors that will ;
* connect the locations. Lower opacity will be more transparent. Full ;
* opacity, 100%, is opaque. Select the opacity by entering the 2-char ;
* hexadecimal equivalent from the 10 choices below. Opacity = 50% was ;
* the default in all previous versions. ;
* ;
* %let trans = '19'; /* 10% opacity */
* %let trans = '32'; /* 20% opacity */
* %let trans = '4b'; /* 30% opacity */
* %let trans = '65'; /* 40% opacity */
* %let trans = '7f'; /* 50% opacity */
* %let trans = '98'; /* 60% opacity */
* %let trans = 'b1'; /* 70% opacity */
* %let trans = 'cb'; /* 80% opacity */
* %let trans = 'e4'; /* 90% opacity */
* %let trans = 'ff'; /* 100% opacity */
*****;

%let trans = '7f';

*****;
* For Google Earth, specify an overall project description that will be ;
* documented in the GE layer title. Then, specify a URL web address ;
* where further information (project details) can be obtained. ;
* If there is no applicable URL, just put a blank space between the two ;
* single quotation marks. ;
*****;

%let ge_species = 'MyProject Description';
%let ge_project_url = 'http://www.myprojectwebsite.org/';

*****;
```

DESCRIPTIONS FOR EACH USER-DEFINED FILTERING PARAMETER:
MINOFFH

Please read carefully; most users misunderstand, and set this parameter wrong!

To assign a correct threshold for this parameter, ask yourself: "What is the minimum amount of time that needs to elapse without receiving any ptt transmissions to ensure that at least one duty cycle has incremented?"

If your ptt did not have a duty cycle (eg. it was on continuously), or if the ptt was "off" for LESS time than it was "on", read the **PICKDAY** variable below first.

The SAS program attempts to group locations based on the ptt duty cycle during which they were obtained. Most ptt's attached to animals have a preprogrammed duty cycle for conserving battery life. For example, 12 hours 'on' followed by 60 hours 'off', or 6 hours 'on' followed by 22 hours 'off'. Therefore, if enough elapsed time has passed without reception of ANY Argos data, then we can assume that AT LEAST ONE duty cycle has elapsed. For example, consider a ptt that was programmed with the duty cycle below, which changes seasonally:

```
Season-1:  6h on / 18h off for  60 days
Season-2:  8h on / 64h off for 150 days
Season-3:  4h on / 96h off for remaining battery life
```

We can conclude that if there was a temporal gap between Argos overpasses (received ptt transmissions) that exceeded 8 hours during any of the 3 seasons above, that AT LEAST ONE duty cycle had transpired.

In the example above, I would suggest a value of 10 be assigned to the **MINOFFH** variable:

```
%let minoffh = 12;
```

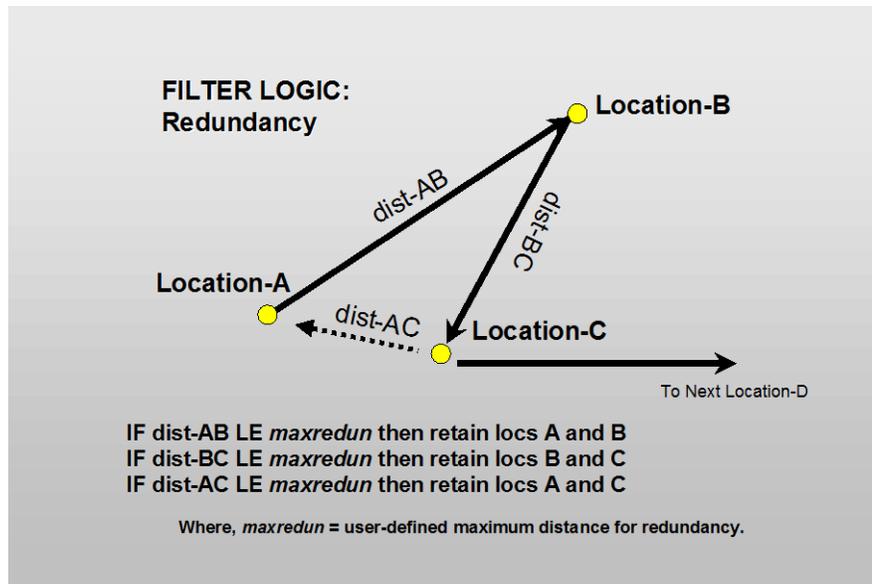
That is, consider your ptt's duty cycle (across all seasons if applicable), ask what is the **MAXIMUM on** duration (units=hours), then add a couple of hours, but don't add so many hours that the value begins to approach the minimum off time (across all seasons). I add some time because some ptt models do not use real clocks, and their estimates of elapsed time vary. I chose 12 hrs in my example above because it will work for any ptt whose duty cycle on-time was <=12 hours and off-time was >12 hours.

Note that if the minimum OFF duration is less than the maximum ON duration, during any season of a ptt's duty cycle, this logic fails. It will not crash the SAS program, but it will mean that the output variables PICK and DUTY will not be valid (see variable definitions below). It's less common that animal ptt's are programmed to transmit with intense battery consumption, but some studies do. If your ptt transmitted continuously (24h/day), then using the 24-hour day would be a logical choice for defining temporal intervals. See the **PICKDAY** variable below.

NOTE: The 'DUTY' variable only increments when a location is actually acquired from Service Argos. The 'DUTY' variable DOES NOT NECESSARILY EQUAL the number of 'real' elapsed duty cycles. When a ptt fails to acquire a location during any given real duty cycle, the 'DUTY' variable does not increment, and thus falls behind the actual number of expired duty cycles.

MAXREDUN

This variable (units=km) is used to filter the Argos locations such that every retained location has a temporally near-consecutive location that is spatially within the defined distance. The premise for such a filter is that larger Argos locational errors rarely occur consecutively in the same place (although it does happen, it is relatively rare in most datasets). The program logic is constructed as follows: The dataset is evaluated chronologically for each animal. Three consecutive locations are evaluated simultaneously - call them locations A, B, and C (see Figure below). The 3 resulting between-location distances are calculated (A->B, B->C, and C->A). If any calculated distance is less than the user-defined threshold (**MAXREDUN**), then both locations (the 2 endpoints of the vector whose length is less than **MAXREDUN**) are retained (they are not filtered). The term 'near-consecutive' is used above, because in the typical situation, location 'B' will be the outlier, and locations 'A' and 'C' will be the pair of redundant, self-confirming locations, even though they are not exactly consecutive. If location-B is flagged for omission, then should it be considered in the next set (triplet) of evaluated locations? Good question - please read the **SKIPLOC** documentation below. If location-B was flagged for filtering but the **SKIPLOC** parameter is set to zero (0), then next set of evaluated locations would be: B->C, C->D, and D->B. If the user-defined variable **KEEPLAST** (see below) is set to a value of one (1), then the last location for each animal will NOT be filtered, regardless of distance to the prior location.



What value should be used for **MAXREDUN**? Very good question. It depends on several factors, but the most important one is 'At what spatial scale are you interpreting your location data?' For example, consider a bird tracked from Canada to Chile (very broad scale). Let's say that during migration, as the bird was travelling over northern regions of South America, 1 Argos location with a Location Class value of 'B' was obtained near the coast of Peru. If this was the ONLY location received after the bird left Cuba and arrived in Chile, then it shouldn't be concluded that this individual used a migratory pathway along the coast of Peru. A single Class-B location could have a very large error. But, what if another location of Class-B quality was obtained that was consecutive to the aforementioned location, and they were within 30km of each other? Now, 2 Argos locations, albeit of low Class-quality, have self-confirmed the same general Peruvian area. Together, they provide a lot more credibility toward concluding that the animal used the coastal region of Peru during migration to Chile. So if you're investigating continental or hemispheric-scale movements, having an animal relocated within 30km is reassuring information.

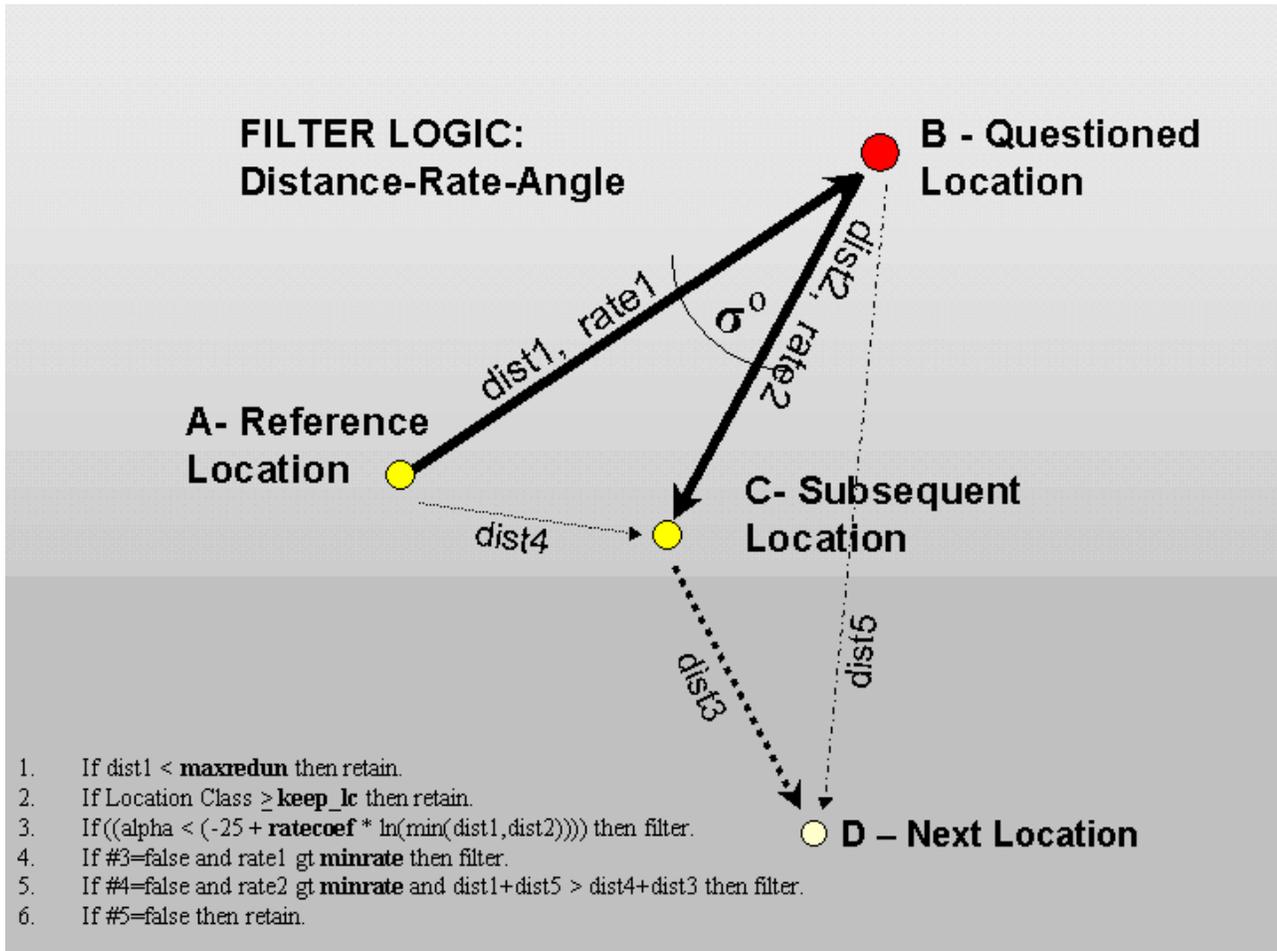
In contrast, imagine a kangaroo study, and that the study animals really didn't go anywhere - they just stayed around an area of +/- 25km. The raw Argos data are probably spread over a large geographic area, a mixture of good and implausible locations. In this situation, a very conservative **MAXREDUN** value of perhaps 2km would be more appropriate. Such conservative values will filter a lot of locations, but it may be worth having a critical look at those locations that are not filtered.

There are other factors involved with the usefulness of the **MAXREDUN** filter, such as duty cycle, latitude, and the real rate of animal movement. For example, I have been using **MAXREDUN** = 10 for filtering polar bear data. The polar bear transmitters typically transmitted for 6 or 8 hours every 3 or 7 days. During their on-period in the northern latitudes, several locations were usually obtained, especially when the pttts were not near the end of their battery life. Because polar bears move relatively slowly, there were enough locations during the duty cycle that were within 10km of one another to provide redundancy, and the more implausible locations were fairly easy to identify. However, if you have a duty cycle that has a relatively short on-time, and/or your study area is near the equator where the frequency of satellite overpasses is minimal, then many or most of your duty cycles may have recovered only a single location. Or, if your animals are moving relatively large distances during (within) the duty cycle, your dataset will contain little locational redundancy, and the redundancy filter may not be very useful. This is typical of avian tracking when the birds are migrating. When birds are at their nesting, staging, or wintering grounds, **MAXREDUN** values of 5km or 10km may be satisfactory, but when the birds are migrating, the distance-angle-rate filter (described below) will be more useful. Merging the results of the redundancy filter and the distance-angle-rate filter, for animals like birds that nest, stage, molt, migrate, and over-winter, is also implemented in this Versions of this Program that are ≥ 5.0 (see **AZMARGIN**, **XMIGRATE**, and **XOVERRIDE** below).

MINRATE

This variable (units= km/hr) is used by the DISTANCE-RATE-AZIMUTH (**DAR**) filter. It represents a measure to check plausibility of an Argos location based on a reasonable rate of movement that your animal might sustain over a period of hours or days. The program logic again assesses 3 consecutive locations simultaneously A, B, and C in chronological order (see Figure below). The program calculates 2 distances and 2 rates-of-movement, based on A->B (dist1 and ratel) and B->C (dist2 and rate2). The program also calculates the angle (alpha, in degrees) that is formed by A->B->C, with B at the vertex.

Location-A is considered a valid reference location (eg. it was not previously selected for omission). Location-B is the location under question. Location C completes the triplet, but it is not considered for omission until the next iteration, when it will become the vertex-location. If location-B is flagged to be filtered, it is not used in subsequent triplets.



The program sequentially evaluates 6 tests (shown in the above figure). Note that location-B will ALWAYS be retained if the distance A->B is less than **MAXREDUN**. In other words, if the animal did not move farther than the user-defined threshold for spatial redundancy, then further tests regarding rate of movement or direction are irrelevant. Two consecutive Argos locations with good accuracy can be associated with an extremely high rate of movement when the elapsed time between the two locations is very small. Note also that if the Location Class for Location-B is greater or equal to the user-defined **KEEP_LC** value, then Location-B is automatically retained. If the user has specified **R_ONLY = 1** (see below), then Test #3 is not conducted.

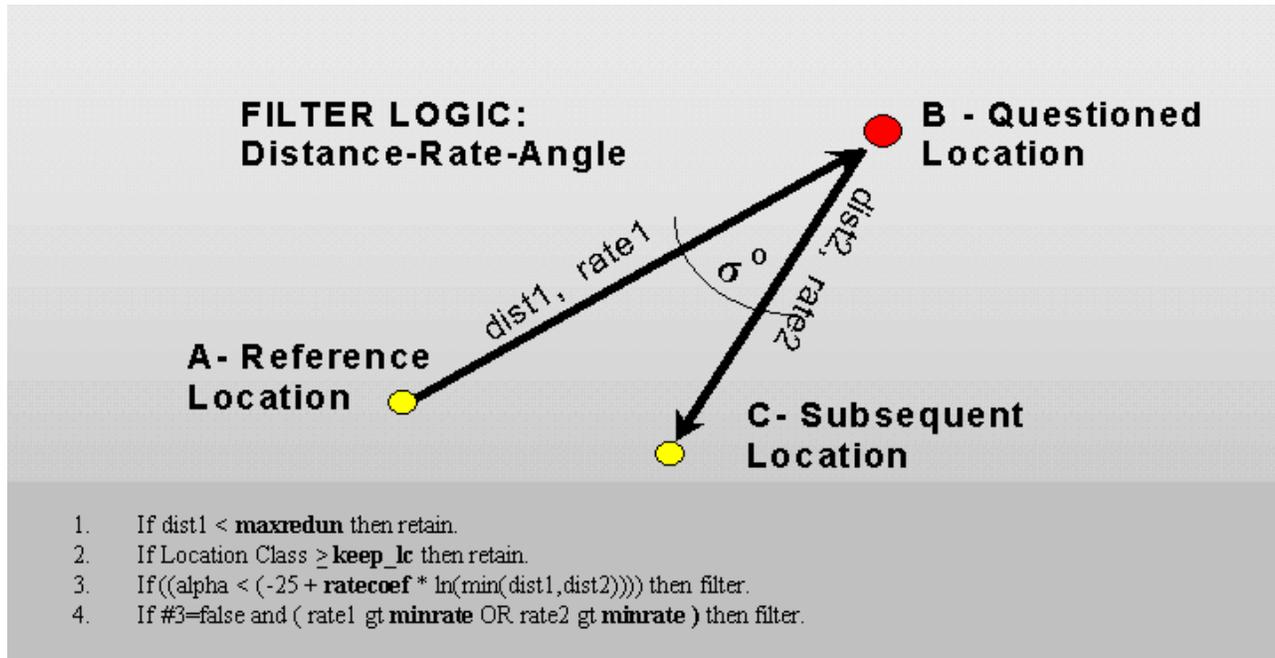
The third step of the assessment evaluates the angle (alpha), and permits the angle to get bigger as the distance of the smaller of the 2 vectors (dist1 or dist2) increases. If the distances are relatively small, a tighter angle is required to cause a filtered result. If the distances are relatively large, then a larger angle will cause implausibility. This is necessary, in part, because the SAS program uses great circle aviation math to derive distance and direction, but it is also biologically reasonable to presume that the farther an animal moves, the less likely it is to have immediately returned back toward the originating region. The user has to define a value for **RATECOEF**, which has a logarithmic influence over the behavior of the angle assessment. **RATECOEF** is discussed in more detail below.

The last part of the assessment requires that at least one of the two vectors has a rate of movement that is suspect (above the user specified minimum **MINRATE**). If $\text{rate1} > \text{minrate}$, then location-B is filtered. If $\text{rate2} > \text{minrate}$ (and $\text{rate1} < \text{minrate}$) then it becomes ambiguous whether Location-B or Location-C (or both) is implausible. Cumulative distances from Location-A to Location-D, either through Location-B or through Location-C, are compared to decide whether Location-B will be filtered.

The entire **DAR** filtering strategy is iterated 3 times, each successive iteration using only locations that passed the previous iteration. It is necessary to iterate the DAR strategy multiple times because

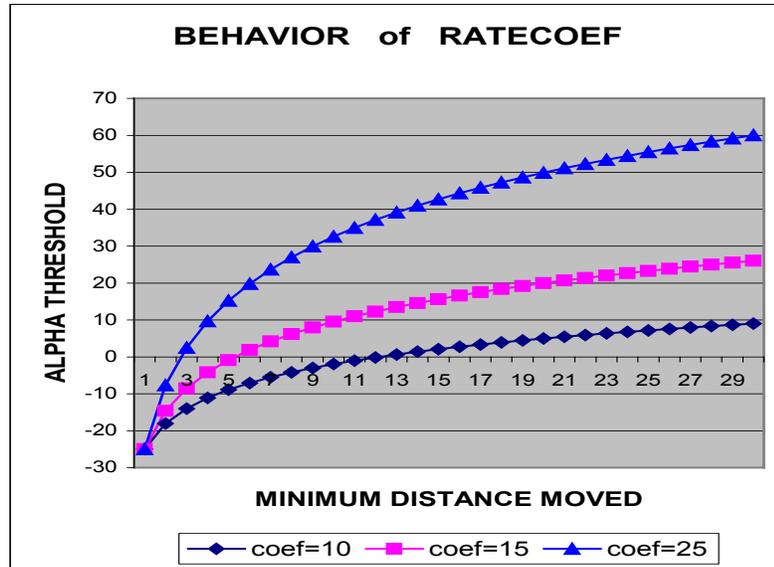
filtering a location during one iteration may create implausible rates of movements or suspicious angles that need to be reevaluated by a subsequent iteration.

Finally, two more **DAR** iterations are performed using a slightly different set of decision rules that are documented in the figure below. These modified rules are identical to the rules used in all previous versions of this program (e.g. Versions 5 and below). In previous program versions, the DAR decisions documented in the figure below were iterated 5 times. In Version 6, slightly different sets of criteria were used in the first 3 iterations to try and improve decision about the fate of Location-B when only rate2 exceeded **MINRATE** (e.g. when it was ambiguous if Location-B, Location-C, or both were implausible).



RATECOEF

This coefficient influences how the angle (α) is evaluated. Larger angles will be considered suspect as **RATECOEF** increases. Use of this coefficient needs further study and empirical testing. The figure below illustrates the behavior of **RATECOEF**. For example, if the shorter of the 2 vectors (A->B and B->C) was 30km and **RATECOEF**=10, then if the angle (α) was less than about 9-degrees, location B would be filtered. If **RATECOEF**=15, then angles less than about 26-degrees would be filtered, and for **RATECOEF**=25, angles less than about 60-degrees would be filtered. Assigning larger values to **RATECOEF** will result in more locations being filtered. More mobile animals should be assigned smaller **RATECOEF** values. I suggest you begin with **RATECOEF**=15 for birds, and **RATECOEF**=25 for mammals and reptiles, then review the filtering results, and adjust **RATECOEF** if you feel it's necessary or you want to experiment. I have used a **RATECOEF** value of 15 for oldsquaw, 20 for swans, and 25 for polar bears, with reasonable results. I experimented with even more aggressive (higher) **RATECOEF** values for some tracking data where the animals moved very little throughout the study. I concluded that distance-angle-rate filtering is less stable, and less intuitive for relatively sedentary animals, and that minimum-redundant-distance filtering is a better choice under such conditions.



```
r_only = 0      (rate only)      Case 1
r_or_a = 1      (rate or angle)  Case 2
```

One and only one choice should be flagged with a value of '1', the other must have a value of zero. The program will immediately abort if more than one choice (or no choice) is made. Filtering is made if:

In Case-1, either rate1 or rate2 must exceed the user defined **MINRATE**, without any regard to the angle formed by the two vectors.

In Case-2, either rate1 or rate2 must exceed the user defined **MINRATE** *OR* the angle formed by A-->B-->C must be suspiciously acute.

Most often, a user will select the 'rate or angle' (Case-2 above). A user might decide otherwise when tracking animals with particular behaviors. For example, consider sea birds such as murres or puffins that frequently travel from a colony to offshore feeding areas, and return. With this type of behavior, we might expect 'out-and-back' movements in the tracking data. The user should consider selecting the 'r_only' choice (rate only), such that locations would only be filtered if the rate of movement exceeded **MINRATE**, regardless of the angles formed. This strategy will likely leave a substantial number of erroneous locations unfiltered. I would suggest careful review of the 'r_only' results, searching for areas of repeated use (eg. look for temporal redundancy across days, or individuals). Repeated use of an area over time, or by multiple individuals serves to confirm, while single 'out-and-back' movements to areas that were never visited again should remain suspect.

GMTOFFST (deviation from GMT, in units=hours)

You may change all date/time values in the output data sets to have values equal to 'local' time, rather than the GMT date/time that Argos disseminates. Unit for the GMT offset variable is hours. For example, tracking conducted in California might set **GMTOFFST** = -8. If **GMTOFFST** is set to zero, then all date/time values are left unchanged (GMT). If you use a value other than zero, it may become problematic to merge or compare the output data sets to the raw Argos data, or to other data summaries. Alternatively, it is easier to interpret diurnal animal behaviors in units resembling local solar time. If your animals moved extensively between time-zones (large east-west movements), it's best to keep **GMTOFFST** = 0.

GMTOFFST does not adjust the DEPLOYMENT Date/Time (dtime). Thus, it is important that the dtime variable (in the deployment logic-blocks) uses the same time zone defined by **GMTOFFST**.

If **GMTOFFST** is set to 0 (zero) then all date/time values in the output data sets will be the same as those in the raw Argos data. If **GMTOFFST** is not equal to zero, the GMT_DATE, GMT_HOUR, GMT_MIN, and GMT_SEC in the output data sets will continue to adhere to GMT time, while all other date/time values will have the offset applied.

```
latmin = -90. ;
latmax = 90. ;
lonmin = -180. ;
lonmax = 180. ;
```

These parameters form a rectangular geographic area-of-validity. Restricting this area of validity can sometimes be useful if you have received very aberrant locations in your raw Argos data. If a location falls outside the geographic rectangle, the record is simply removed from all outputs.

KEEP_IC

This variable must be set to either 3, 2, 1, any other value will cause the program to fail. This parameter establishes the minimum Argos Location Class that will NEVER be filtered by either the Minimum Redundant Distance (**MRD**) Filter or the Rate Angle Distance (**DAR**) filter. The value you choose should be commensurate with the spatial scale of data analysis. We have very rarely observed an Argos Location Class 3 to have accuracy worse than about 1km (usually they are within 0.5km). However, if you are studying continental or broad regional movements, then the larger location errors associated with LC2 and LC1 (+/- 5 or 10 km respectively) may be negligible compared to the overall movements you are studying. If this is the case, then you should consider keeping all LC1, LC2 and LC3 locations (prevent them from being filtered) - thus set **KEEP_IC** = 1. If you are studying more local scale movements, set the **KEEP_IC** value to 3, and let redundancy self-validate the Argos data. If you are studying broad-scale movements, use **KEEP_IC** = 1.

RANKMETH

This variable allows the user to choose between 3 different equations that determine which location (within a duty cycle) will be identified as the 'highest quality', and carried forward into the 'Pick' (pk) datasets. The 'highest quality' location per duty-cycle is chosen based on the Argos Location Class (3,2,1,0,A,B,Z), and in the event of a tie, then the **RANKMETH** value will determine which equation is implemented to resolve the ambiguity. Valid **RANKMETH** values are 1, 2, or 3, where the value identifies one of three equations to resolve which location will be selected for the 'pick' datasets, GIVEN that the LC values are a TIE:

rankmeth = 1: The location with the highest IQX will prevail, else if a tie, the location with the highest IQY will prevail, else if a tie, the location with the most number of messages. This is equivalent to the method used in Versions 6.0-6.4.

rankmeth = 2: The location with the highest IQX will prevail, else if a tie, the location with the most number of messages will prevail, else if a tie, the location with the highest IQY. I personally lean toward this option, at this time. If you feel otherwise, I'd like to hear.

rankmeth = 3: The location with the most number of messages will prevail. This is equivalent to the method used in Versions 5 and earlier.

KEEPLAST

Set this variable to 1 (one) if you want to retain the last location for each animal, within the minimum-redundant-distance filter, regardless of its location. Set the variable **KEEPLAST** to zero if you want the last location for each animal to be filtered if it is greater than **MAXREDUN** from the previous location. Recall that the last location for each animal is NEVER filtered by the distance-angle-rate (**DAR**) method.

PICKDAY

Set this variable to 1 (one) if you want to select the location with the best Argos location class based on 24-hour periods instead of the ptt's duty cycle. While duty-cycle is typically a better interval from which to pick the 'best' Argos location, sometimes a biologist builds ptt's that do not have duty-cycles, e.g. the ptt is on continuously. In this case, by setting **PICKDAY** = 1, the location with the best Argos Class will be selected based 24-hour periods (eg. daily).

Important Note: If you use **PICKDAY** = 1, then set **MINOFFH** = 0.00001;

SKIPLOC

Consider 5 consecutive locations, and call them A, B, C, D, and E. The minimum-redundant-distance-filter considers consecutive triplets of locations; the first triplet would then be A, B, and C. Three

distances are calculated: AB, BC, and AC. If any of the 3 distances are less than **MAXREDUN**, the two respective end-point locations are retained by the filter. A question arises in the event that location 'B' is rejected: should location 'B' be considered in the next triplet? If **SKIPLOC=0**, then location 'B' is considered, and the next triplet would be B, C, D; and the 3 distances BC, CD, AND BD would be evaluated. If **SKIPLOC=1**, then location 'B' is not considered (it is skipped), and the next triplet would be C, D, E; and the 3 distances CD, DE, and CE would be evaluated. Most feedback I've had indicates that users prefer **SKIPLOC=0**.

XMIGRATE, XOVERRUN

These variables are used to combine the minimum redundant distance filter results (**MRD**) with the distance angle rate filter results (**DAR**). When an animal is actively migrating, locational redundancy is difficult or impossible to obtain. Often, locations during active migration only pass the **DAR** filter. Beginning with Version 4.0 of this program, datasets are created that pool results from both filters. The logic assumes the **MRD** results are superior, and considers those locations that passed the **MRD** filter to be 'anchor' points. Ultimately, the merging routine supplements the **MRD** results with locations that passed the **DAR** filter, ONLY IF the **DAR** locations pass a user-defined number of directional movement tests (described in the next section). The user-defined variables **XMIGRATE**, and **XOVERRUN** control whether a **MRD** vector (a vector based on two consecutive **MRD** anchor locations) represents a 'Migration Event'. If it represents a migration event, intervening **DAR** locations collected during that vector-period are evaluated and potentially amended to the output data set, if those intervening **DAR** locations adhere to the directional movement defined by the **MRD** vector. The 'Migration Event' test for a **MRD**-vector is passed if:

1. The **MRD**-vector length exceeds **XMIGRATE*MAXREDUN**; and
2. The intervening **DAR** vector does not exceed the **MRD** vector length + **XOVERRUN*MAXREDUN**

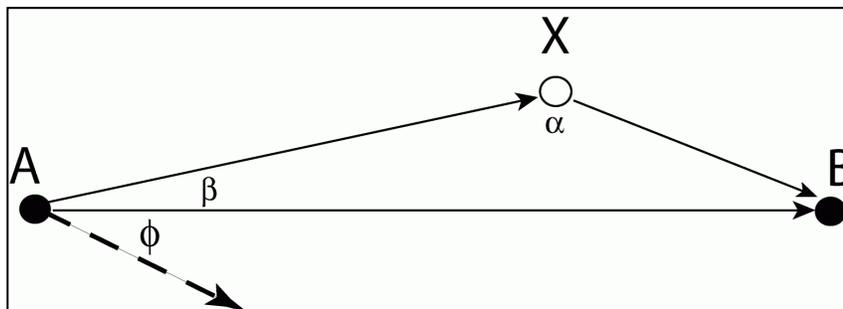
Thus, intervening **DAR** locations are not considered if the distance between two consecutive **MRD** locations was not representative of a 'Migration Event'. The distance defining a migration event is calculated as: **XMIGRATE*MAXREDUN**. Therefore, if you had defined a relative small value for **MAXREDUN**, then the **XMIGRATE** factor would have to be adjusted larger in order to maintain the same threshold for defining a migratory movement.

The parameter **XOVERRUN** prevents a **DAR** location from being retained if that location results in a vector which exceeds the distance of the original **MRD** vector by an amount of **XOVERRUN*MAXREDUN**. In other words, it attempts to prevent a bird from over-flying the **MRD** endpoint 'anchor' destination.

XDIRECT, XANGLE, XPERCENT

These variables are used when combining the minimum-redundant-distance filter results (**MRD**) with the distance-angle-rate filter results (**DAR**). These 3 parameters are used to make 3 independent tests of directionality. Using the **MRD** locations as anchor points, EACH intervening **DAR** location is INDEPENDENTLY assessed, as described below. Assuming that the **XMIGRATE** and **XOVERRUN** tests (above) were passed, 3 directionality tests are conducted: DIRECTION, ANGLE, and PERCENT.

In the figure below, consider two consecutive locations (A and B) that passed the **MRD** filter, and therefore have become anchor points. An intervening location (X) had passed the **DAR** filter, and will now be considered for amendment to the **MRD** results. Acceptance or rejection is based on results of three directionality tests.



TEST #1 - DIRECTION:

If the azimuth of A->X is within +/- **XDIRECT**-degrees (a user-defined threshold), location X passes the direction test. Units for **XDIRECT** are degrees, and the threshold in the figure above is illustrated by the angle ϕ . In other words, if the azimuth of the vector A->X is within +/- **XDIRECT** degrees of the azimuth of A->B, then the direction test is passed.

TEST #2 - ANGLE:

This test evaluates the angle formed by A->X->B (angle α in the figure above). If **XANGLE** is greater than or equal to α , location X passes the angle test. **XANGLE** should not exceed 180.

TEST #3 - PERCENT DISTANCE:

If the total distance of travel along A->X->B is no greater than a user-defined percent (**XPERCENT**) of the anchor-distance A->B, then X passes the PERCENT DISTANCE test. That is, how much farther are you willing to let your animal 'wander' in order to pass through location X on its way to location B: 10%, 25%, 50%?

TESTP_OA & TESTP_BZ

TESTPASS defines how many of the 3 tests above (1, 2, or 3) must be passed in order for the location in question (X in the Figure above) to be retained in the 'Best Hybrid' output data set. If **TESTPASS** is defined as '1' (the most liberal), then only one (any one) of the three tests is required to pass. If **TESTPASS** is set equal to 2, then any two of the 3 tests must pass, and if **TESTPASS** is set equal to three (the most conservative), all 3 tests must pass.

Beginning in Version 6.5, **TESTPASS** was partitioned into two parts: 1) cases where the location in question is an LC=Z or LC=B; and cases where the location in question was an LC=A, LC=0, or better. If both **TESTP_Oa** and **TESTP_Bz** are set to the same value, then the net result is no different than the **TESTPASS** parameter in versions preceding 6.5.

If the user wants to be more conservative with LC=Z and LC=B locations, then the user might assign, for example:

```
%let testp_Oa = 2 ;
%let testp_Bz = 3 ;
```

The above requires that LC B and Z locations pass all three directionality tests in order to be retained in the 'Best Hybrid' output, while LC A and 0 locations need only pass 2 directionality tests.

If the user has defined **KEEP_LC** = 2 or **KEEP_LC** = 3, then the **TESTP_Oa** threshold also applies to number of affirmative directionality tests of any intervening LC1, or any LC1 and LC2 locations, respectively.

GOOGLE EARTH DATE ANNOTATION TEXT SCALE

Each location in Google Earth is annotated with the Date/Hour. Size to the text annotations can be varied using by increasing or decreasing the scaling parameter "txtscale". The date of the most recent location "txtscalelast" can be emphasized by making its scale larger. This is strategy of the values used below. Experiment to attain the text scaling you prefer.

```
%let txtscale = 1.0;
%let txtscalelast = 1.5;
```

OUTPUT VARIABLE DEFINITIONS:

The table below defines each of the variables that appear in the ASCII output files. Not all variables appear in each of the ASCII files that are created. "(DIAG)" denotes a parameter that is directly copied from the original Argos DIAG input data. Other variables are present, but not defined below; hopefully they are self explanatory.

Name	Format	DESCRIPTION
altitude	Num	Argos ptt reference altitude (DIAG)
animal	Char	User defined Animal ID
apbest	Num	Best signal strength during pass (DIAG)
apfreq	Num	Interpolated freq at Doppler inflection (DIAG)
azimuth	Num	Great-circle compass bearing (degrees) to next loc
azfrmdply	Num	Great-circle compass bearing (degrees) from the deployment location
bothtest	Char	Which 3 directionality tests passed in 'Best Hybrid'
date	MMDDYY	Date of overpass/location (GMT + (GMTOFFSET))
day	Num	Calendar day of overpass/location (GMT + (GMTOFFSET))
daysdply	Num	Number of elapsed days since deployment date
ddmm	Char	DAY MONTH character string
deltahr	Num	Number of elapsed hours since previous pass
deployed	Char	Character date of deployment
deplylat	Num	User declared deployment latitude
deplylon	Num	User declared deployment longitude
distance	Num	Great-circle distance (km) from previous location
dist2dply	Num	Great-circle distance (km) from deployment location
dur	Num	Duration (sec) between first and last message (DIAG)
duty	Num	Unique number for all records within a duty cycle
dutylocs	Num	Total number of passes/locations within duty cycle
first	Num	1/0 flag for first data record for a given animal
gmt date	MMDDYY	GMT date appearing in the raw Argos data (DIAG)
gmt hour	Num	GMT hour appearing in the raw Argos data (DIAG)
gmt min	Num	GMT minute appearing in the raw Argos data (DIAG)
gmt sec	Num	GMT second appearing in the raw Argos data (DIAG)
gt120	Num	N messages with signal strength <120dB (DIAG)
hour	Num	Hour of overpass/location (GMT + (GMTOFFSET))
iqx	Num	IQX value (DIAG)
iqy	Num	IQY value (DIAG)
jd	Num	Julian day YYDDD (GMT + (GMTOFFSET))
jday	Num	Julian day DDD (GMT + (GMTOFFSET))
latitude	Num	Chosen lat (1° vs 2°) based on minimum distance
lat a	Num	Alternate latitude (DIAG)
lat p	Num	Primary latitude (DIAG)
latest	Num	1/0 flag for most recent data record for an animal
lc94	Char	Argos location class
line	Char	Unique LINE-Identifier (ANIMAL-ID MMDDYYYY : HHMMSS Z99)
longitud	Num	Chosen lon (1° vs 2°) based on minimum distance
lon a	Num	Alternate longitude (DIAG)
lon p	Num	Primary longitude (DIAG)
minute	Num	Minute of overpass/location (GMT) (DIAG)
mmdd	Char	MONTH DAY character string
month	Num	Month of overpass/location (GMT + (GMTOFFSET)) (DIAG)
nlocrank	Num	Weighting for best LC94 within duty cycle
nmess	Num	Number of messages received during overpass (DIAG)
nopc	Num	Number of plausibility checks passed (DIAG)
pick	Num	1/0 flag denoting best LC94 within duty cycle
ptt	Num	Ptt number (DIAG)
rate	Num	Rate of movement (km/hr) from previous loc
sampsize	Num	Total number of records for respective animal
second	Num	Second of overpass/location (GMT) (DIAG)
sen1	Char	First sensor entry from first line of data (DIAG)
sen2	Char	Second sensor entry from first line of data (DIAG)
sen3	Char	Third sensor entry from first line of data (DIAG)
sen4	Char	Fourth sensor entry from first line of data (DIAG)
switch	Num	Location Choice: 0=primary 1=alternate 3=primary 4=alternate
weeddist	Num	1/0 flag 1= reject based on minimum redundancy MRD
weedrate	Num	1/0 flag 1= reject based on distance-angle-rate DAR
weebboth	Num	1/0 flag 1= reject amending DAR loc with MRD anchors
year	Num	Year of overpass/location (GMT + (GMTOFFSET)) (DIAG)

OUTPUT FILE DEFINITIONS:

The SAS program automatically generates the "filename" and embeds 3 of the more critical user-defined filtering parameters (numbers), so results derived with different parameter choices can be later compared.

d# - your choice for the minimum-redundant-distance (km)
 r# - your choice for maximum sustainable rate of movement (km/hr)
 lc# - your choice for the best Argos location class to ALWAYS retain

The ASCII files (found in the .\ascii subdir) are named by prefixing with the project_name and then with the following conventions, which contain:

- tottot.txt** - One record for every DIAG overpass, regardless if a location was reported. This file provides a complete dataset of ALL Argos records. It is suited to studying ptt performance; it is not applicable for GIS since location data may be missing for many records.
- totloc_d#_r#_lc#.txt** - One record for every DIAG overpass that fixed a location. Each location record is flagged with three pass-fail 1/0 variables, one for each filtering method.
- d#lc#.txt** - Subset of totloc*.txt which contains ONLY the locations that passed the minimum-redundant-distance (**MRD**) plausibility filter - plus ALL Class-lc# and above locations. This subset contains NEW calculations of DISTANCE, RATE and AZIMUTH variables.
- r#d#lc#.txt** - Subset of totloc*.txt which contains ONLY the locations that passed the distance, angle, rate (**DAR**) plausibility filter - plus ALL Class-lc# and above locations. This subset contains NEW calculations of DISTANCE, RATE and AZIMUTH variables.
- br#d#lc#.txt** - Locations in d#lc# PLUS additional locations from the r#d#lc# data set that conformed to the overall directional movement defined by the locations in dt#lc#. This is the pooled **MRD** and **DAR** dataset (the "best" hybrid). A few of the original **MRD** locations (anchor points) may be removed because a final movement-rate filter is applied to this data set.
- pkd#lc#.txt** - Subset of d#lc#.txt which contains ONLY the locations that were selected as the 'highest quality of the duty-cycle'. The 'highest quality' location per duty-cycle is chosen based on the Argos Location Class (3,2,1,0,A,B,Z), and in the event of a tie, the RANKMETH user-defined variable (see below) will determine which location is chosen. The "pick" (pk*) subsets contain NEW calculations of the DISTANCE, RATE and AZIMUTH variables.
- pkrd#lc#.txt** - Subset of r#d#lc#.txt which contains ONLY the locations that were selected as the 'highest quality of the duty-cycle'. The 'highest quality' location per duty-cycle is chosen based on the Argos Location Class, as described above.
- pkbr#d#lc#.txt** - Subset of br#d#lc#.txt which contains ONLY the locations that were selected as the 'highest quality of the duty-cycle'. The 'highest quality' location per duty-cycle is chosen based on the Argos Location Class, as described above.

NOTE: It is the user's decision which filtering method (e.g. output dataset) best suits their analytical requirements. See the 2012 MEE paper. The **MRD** filter is often the best choice if tracking a slow moving animal like a mammal. Avian tracking during active migration poses challenges to the **MRD** filter because spatial redundancy is very difficult to obtain. The **DAR** filter often performs better during active migration, while the **MRD** filter is thought to be better during sedentary periods such as nesting, staging, molting, and migratory stopovers. The "best" hybrid filter is an attempt to maintain the robustness of the **MRD** filter during sedentary periods, and use **DAR** results only during migratory periods. Keep in mind that **DAR** locations retained in the "best hybrid" (**HYB**) data sets are those that passed directional movement tests. Consequently, we have imposed a constraint (bias) that the animal migrates in a more or less 'straight line'. But, how correct is this constraint? The 3 respective output subsets (files beginning with the letter 'p') contain just one location per duty cycle (or one location per 24h if the ptt is continuously on), and these datasets are best used for spatial analyses like home range because they reduce problems associated with temporal autocorrelation.

THINGS YOU SHOULD KNOW:

01. The program never filters Argos Class-3 locations. If the user defined variable **KEEP_LC** is set to 2 (two), then the program never filters Class-2 or Class-3 locations. If **KEEP_LC** is set to 1 (one), then the program never filters Class-1,2, or 3 locations. If the user defines **MAXREDUN** ≥ 10 km, then it would make sense to set **KEEP_LC** = 1, since Class 1, 2, and 3 locations usually have accuracy within 10 km (e.g. there is no reason to filter the Standard Location Class data).
02. In the **DISTANCE-ANGLE-RATE (DAR)** filter, the last location for each animal is NEVER filtered. In the **MINIMUM-REDUNDANT-DISTANCE (MRD)** filter, the last location for each animal may or may not be filtered, depending on its proximity to the previous location and the value you assign to the **KEEPLAST** parameter.
03. If an animal has ONLY 1 or 2 locations in the entire data set, no filtering is attempted.
04. If you are interested in a detailed analysis of ptt sensor data, this is not the program to use. The Argos **DISPOSE** file format contains your full set of sensor data; the **DIAG** format only contains ONE message of sensor data per overpass which simply serves as a solitary 'example' of your sensor data.
05. If you study the SAS program code, you should know that goodlat=latitude and goodlon=longitud. Most of the program code references goodlat/goodlon, but I changed the variable names at the end of the program to accommodate default entry into ARC-GIS through **TABLES**.
06. Don't be alarmed by the series of 'INVALID DATA' errors that will appear in the SAS Log after reading in the input **DIAG** data file. The errors arise when the program tries to read a lat/lon in a **DIAG** overpass that only has '????' reported the lat/lon fields (No Argos location was reported). In this case, lat and lon are set to SAS missing values, which is correct.
07. **CAUTION:** The output ASCII files contain ONE **DEPLOYMENT** record for each animal. This was necessary to correctly choose and filter the very first **ARGOS** location. However, most variables for this deployment record are invalid, they are set to dummy values or they are just duplicate values of the first **REAL** Argos data for the respective animal. Before quantifying metrics about the **ARGOS** data, be sure to remove the deployment records. In the deployment records, the variable **LC94** equals 'DP', so you can use this variable to selectively remove the deployment records.
08. Two files in **HTML** (web-browser) format are written to the **.\ascii** directory. One contains documentation of the user-defined filtering parameters and the other contains frequency tables of filtered and unfiltered locations, within Location Class categories, and **BY PTT**. Double click on these files to view them in your default web-browser.
09. The choice of the best quality Argos location per duty cycle (variable 'pick') may vary between the output ASCII files that underwent filtering, because the location that was picked in one data set may have been filtered in another data set, e.g. the variable 'pick' is chosen independently for each data set.
10. You may find it necessary to 'override' the **DAF**'s choice between the primary and alternate Argos location for some specific overpasses. This situation happens more frequently when an animal is moving large distances between duty cycles, and/or the ptt is near the end of its battery life. There is an accompanying ASCII text file named "hardcode_changes.txt" in which you can:
 - 1) force the primary location to be chosen;
 - 2) force the alternate location to be chosen;
 - or 3) remove the entire overpass from all output data sets.The "hardcode_changes.txt" file contains further documentation. Obviously, you need to study the initial filtered results in a **GIS** or **GE** to discover any aberrant overpasses that would benefit by forcing the choice between the primary and alternation location (or removing the overpass entirely). The examples I provide are commented, you must 'uncomment' the SAS code by removing the preceding * (asterisk) in column #1 of each line. Also, be very careful when you edit the "hardcode_changes.txt" file - **ONLY USE** an ASCII editor, and **DO NOT CHANGE** the file's name: it must be named "hardcode_changes.txt". You can copy/paste the **LINE ID** record from Google Earth into the **hardcode_changes.txt** file. Just click on the location-icon in Google Earth to display the **LINE ID**, and several other attributes associated with the respective location.

On the next page are examples of 3 strategies using logic blocks to (respectively): 1) force selection of the alternate Argos location; 2) force selection of the Argos primary location; and 3) completely remove the overpass from all output data sets.

```
if line = 'bhg0101162000:224848Z99' then do;
  lat_p = lat_a;
  lon_p = lon_a;
  file log; put 'Forced alternate location at: ' line= ;
end;

if line = 'bhg0303182000:201439Z99' then do;
  lat_a = lat_p;
  lon_a = lon_p;
  file log; put 'Forced primary location at: ' line= ;
end;

if line = 'bhg0303182000:201439Z99' then do;
  return;
  file log; put 'Removed an entire Argos record at: ' line= ;
end;
```

My contact information, should you have any questions or suggestions:

David Douglas
USGS Alaska Science Center
250 Egan Drive
Juneau, AK 99801

907-364-1576 voice

ddouglas@usgs.gov

APPENDIX A: Configuring the TELNET (live update) capability of DAV v8.50

Configuring the SAS program to conduct Telnet sessions with CLS is very easy. Modify the `get_argosdata_telnet.sas` program that is in the `.\sas` subdirectory. These instructions are also embedded as comments directly within the SAS program file.

```
*****;
* YOU MUST: EDIT THE 10 %let STATEMENTS BELOW. ONLY CHANGE THE PARAMETERS ON THE RIGHT ;
* SIDE OF THE EQUAL SIGN. THE PARAMETERS ARE DEFINED 1-10 BELOW. ;
*****;
*
* 1. Directory where you want the DIAG data file to be written. (in quotes) ;
* 2. Directory where you want the DISPOSE data file to be written. (in quotes) ;
* 3. Directory where TELNET PROGRAM resides. (unquoted) ;
* 4. Directory where TELNET PROGRAM resides, again. (retain double quotation marks) ;
* 5. The Argos Program Number (numeric only) ;
* 6. The Argos Program Username (retain the single quotation marks) ;
* 7. The Argos Program Password (retain the single quotation marks) ;
* 8. Get Argos data for how many previous days? (numeric, maximum = 9) ;
* 9. Number of hours to add to your PC's clock to achieve UTC time? (range +/- 12) ;
* 10. Number of hours from present to exclude partially processed Argos data (suggest 4+) ;
* ;
*****;

%let diagdir = 'd:\argos\projects\template\rawdiag';
%let dispdir = 'd:\argos\projects\template\rawdisp';
%let prgmdir = d:\argos\projects\template\telnet_util;
%let qprgdir = "d:\argos\projects\template\telnet_util";

%let argpgm = 12345;
%let user = 'ARGOSUSER';
%let pass = 'ARGOSPASSWORD';

%let lagdays = 9;
%let gmthrs = 9;
%let delayhrs = 4;

*****;
1. You can copy/paste all SAS code from the line %let argpgm= to the end, and tailor the second
block of code to Telnet data from a different Argos Program. This is essential if your PTTs
of interest are spread among more than one Argos Program.
2. You should balance the frequency that you run this program with the lagdays value. That is,
if you intend to schedule the Telnet download to occur daily, then setting lagdays=3 may be
a good balance. In this example, each download would be expected to have 2 days of overlap
with the previous download. Having overlap is useful so you don't introduce data holidays
in the event that this automation encounters a problem (for example, a power outage or
interruption in internet connectivity. Downloading weekly should be accompanied by
lagdays=9. Downloading daily and setting lagdays=9 will result in a lot of data redundancy.
Think long and hard and clearly about how you schedule. Have a backup plan. Just because
you set up this Telnet capability in a Windows Scheduler doesn't mean that it will work
every time -- while you go on a 3-week vacation. Someone should be monitoring -- so you
don't introduce data holidays into your archive. Computers (especially Windows) sometimes
hang or crash. If you have 2 computers with SAS licenses, you can run the DAF on two
computers to add a full second-level of system redundancy.
3. If you want your updates to contain the very very most recent data, you need to set
delayhrs=0. If obtaining the very most recent data is not mission critical, it is advisable
to set delayhrs=4 or greater. This is because the very very most recent Argos locations may
change during the course of 4-8 hours as CLS obtains a full complement of transmission data
from both real-time and tape-playback satellite communications. This phenomenon can
introduce locations with slightly different time stamps (by a few minutes or seconds) that
appear to have location redundancy, but they are not independent location estimates. The
DAF attempts to ID and remove these instances, but the method is not 100% accurate. So, by
setting delayhrs=8, the locations you download will be stable, and won't change when you
download the next time (given the temporal overlap you've built into the download schedule).
```

Finally, you must ensure that the `%include` statement in the main `00_template_runv850.sas` program that executes the Telnet SAS program is NOT commented-out (no leading asterisk):

```
filename dotelnet "&dletter.\argos\projects\&pname.\sas\get_argosdata_telnet.sas";
%include dotelnet;
```

APPENDIX B: Configuring Windows Scheduler to run DAF on a defined schedule (automated live update)

This section contains a basic Scheduling "how to" for Windows-7. The same capability is available in Window-XP, but the XP menus are notably different (albeit a bit easier and more intuitive). If you use XP, you should be able to study these instructions, and adapt them to the XP menus.

I strongly recommend you entirely read these instructions before beginning. You definitely want to compose the entries for the "Actions Tab" (see below) in advance (put them in a NotePad text file, so you can copy/paste them into the "Actions Tab" when the time comes).

First, open the Task Scheduler: START->ALL PROGRAMS->ACCESSORIES->SYSTEM TOOLS->Task Scheduler [or search the word schedule in the Start Search box]

On the right-side panel labeled Actions, choose "Create Task..."

Under the "General Tab" (shown below), in "Name", type a meaningful name for this specific task. It could be anything useful; it simply becomes an entry in your Scheduled Task Table. Leave the rest as the defaults. Your task may or may not work if you choose to "Run whether user is logged on or not", as this option presents security risks that some IT administrators block. The downside is that for your jobs to run as scheduled, you (your user login) must remain logged into Windows 24/7.

The screenshot shows the 'Create Task' dialog box in Windows Task Scheduler, with the 'General' tab selected. The 'Name' field contains 'Argos update for TEMPLATE'. The 'Location' field is empty. The 'Author' field contains 'DDOUGLAS-UNIV\ddouglas'. The 'Description' field is empty. Under 'Security options', the 'When running the task, use the following user account:' section shows 'DDOUGLAS-UNIV\ddouglas' with a 'Change User...' button. The 'Run only when user is logged on' radio button is selected. There are also checkboxes for 'Do not store password. The task will only have access to local computer resources.' and 'Run with highest privileges'. At the bottom, there is a 'Hidden' checkbox and a 'Configure for:' dropdown menu set to 'Windows Vista™, Windows Server™ 2008'. 'OK' and 'Cancel' buttons are at the bottom right.

Under the "Triggers Tab" (shown below), choose "New", then select the scheduled frequency. In this example, I've chosen every other day, at 5:00AM; and I stop the task if it runs longer than 30 minutes. Unless you have a slow computer, or a ton of data, 30 minutes should be ample time to download new data and run all data through the DAF. Make sure the "Enable" box is checked.

Begin the task: On a schedule

Settings

One time Start: 11/ 3/2012 5:00:00 AM Synchronize across time zones
 Daily Recur every: 2 days
 Weekly
 Monthly

Advanced settings

Delay task for up to (random delay): 1 hour
 Repeat task every: 1 hour for a duration of: 1 day
 Stop all running tasks at end of repetition duration
 Stop task if it runs longer than: 30 minutes
 Expire: 11/ 3/2013 8:52:37 AM Synchronize across time zones
 Enabled

Under the "Actions Tab" (shown below), there are 3 boxes that must be populated very accurately.

Program/Script: the full path to your sas.exe program. The executable SAS program file resides in the directory where SAS was originally installed. On my computer, this happens to be:

`C:\Program Files\SAS\SASFoundation\9.2\sas.exe`

Add Arguments: full path to the SAS configuration file, and full path to your '00_run' SAS program; all on ONE long line. For my example, this is (take care that quotation syntax is not changed):

```
-CONFIG "C:\Program Files\SAS\SASFoundation\9.2\nls\en\SASV9.CFG" -SYSIN  
"D:\argos\projects\template\sas\00_template_runv850.sas"
```

Start In: full path to the projects .\sas directory. In my example, this is:

`D:\argos\projects\template\sas`

You must specify what action this task will perform.

Action: Start a program

Settings

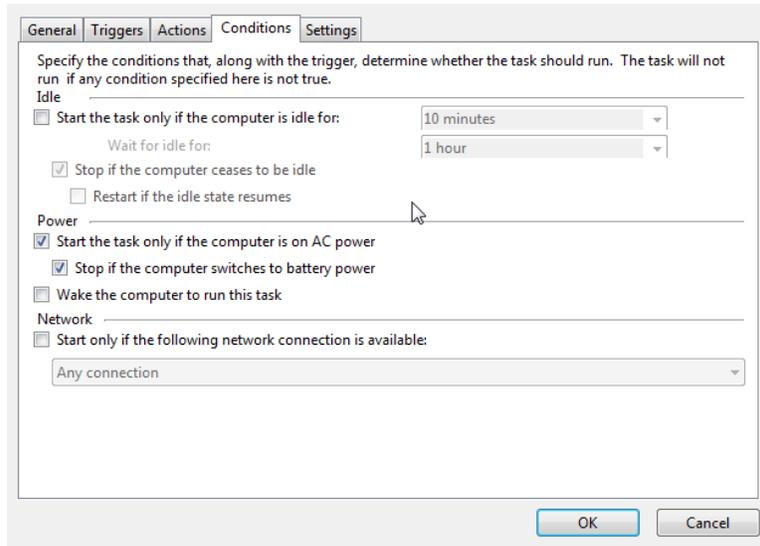
Program/script:
C:\Program Files\SAS\SASFoundation\9.2\sas.exe Browse...

Add arguments (optional):
{-CONFIG "C:\Program Fi

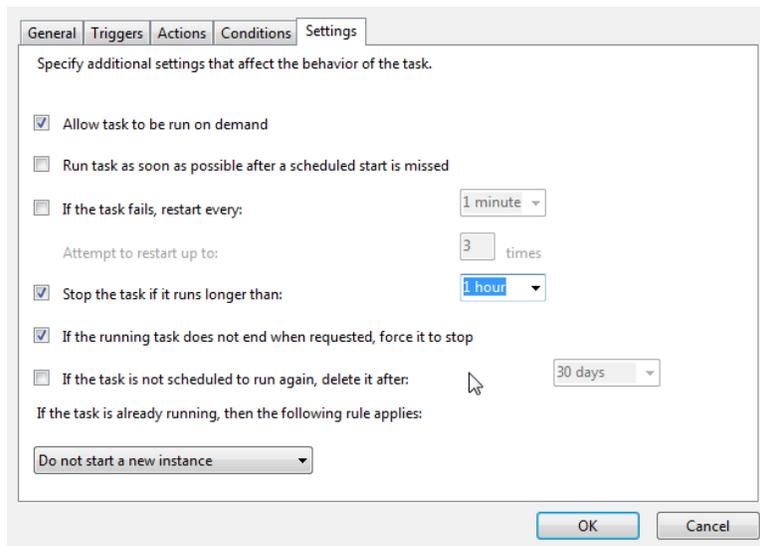
Start in (optional):
D:\argos\projects\templ

OK Cancel

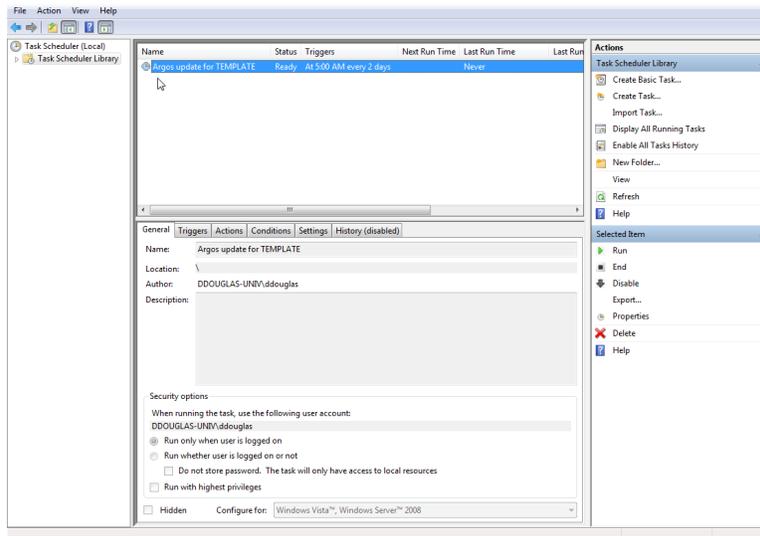
Under the "Conditions Tab", leave the defaults.



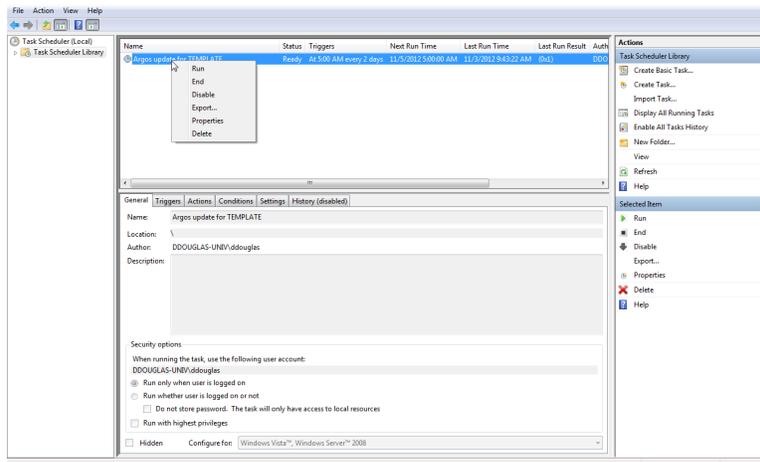
Under the "Settings Tab", reduce the run-time threshold to 1-hour; leave the rest as defaults.



Now, select OK, and this will put the Task Entry into the "Task Scheduler Library". Then, click on "Task Scheduler Library" on the left side, then click on the "Task Name" in the library contents. When the task is "selected" in this manner, it can be edited in any manner (corrections, change the schedule, manually run, delete, etc.).



Assuming the task is 'Enabled' and has 'Ready Status', you should be able to manually run the task by right-clicking on the task-name and choosing 'Run'. If successful, then new filtered output files will appear in the project directories, and a SAS log and listing file will be written to the .\sas subdirectory. If the 'Run' is successful, it is very likely that the program will run in the future at the scheduled frequency.



APPENDIX C: Configuring EMAIL capability within DAF v8.50 (automated update distribution)

All files that need to be modified are located in the `.\email_latest` subdirectory.

File: `argos_email_commands.txt`

This file runs a software utility named 'blat', which basically sends the email. The `.\email_latest` subdirectory contains a user manual for blat, if you are interested.

Open the `argos_email_commands.txt` in an ASCII editor, such as NotePad. Study the blat command line. Contents of the distribution example for this file are shown below. REMOVE and REM characters in columns 1-3 (Remarks). In the distribution file, the commands are Remarked (email is disabled).

This configuration is setup to use GMAIL as the SMTP server, which means it must use a secure shell, which means I have previously installed the public-domain software utility called 'stunnel' (see below). If you have access to an SMTP server that does not require a secure shell, then you should REMARK the first two lines by inserting the letters REM <space>, starting in column #1.

If you want to use the GMAIL SMTP server, then you must first install stunnel (see below). Then, in the `argos_email_commands.txt` file, modify the first 'blat' statement to have YOUR gmail user email address and password (denoted in red font). In my example, the GMAIL user is `douglasargosfilter`, and the password is `sattrack`. DO NOT change the second line that begins with 'set smtpServer...'. In the third line (the second blat statement), change/adapt the entries in bold red font.

```
blat -install smtp.gmail.com douglasargosfilter@gmail.com -u douglasargosfilter -pw mypassword - - gmailsmtp
set smtpServer=-server 127.0.0.1:1099
```

```
blat latest_table2.html -tf my_recipients.txt -server 127.0.0.1:1099 -f douglasargosfilter@gmail.com -u
douglasargosfilter@gmail.com -pw mypassword -debug -log email_log.txt -timestamp -af my_attach.txt -s "ARGOS
PROJECT tracking update"
```

You will also need to modify the 'my_recipients.txt' file and the 'my_attach.txt' file. As indicated by the filename, enter the email addresses of recipients, and the files you want to attach to the email, in these respective files. Note that the attachment filenames use relative pathnames, with the `.\email_latest` subdirectory as the origin directory.

Finally, you will want to modify the 'my_note.txt' file that is in the `.\email_latest` subdirectory. DO NOT change the filename -- the file must exist, although it can be empty if you prefer. Contents of `my_note.txt` are embedded at the end of the email main body.

To install stunnel, see the next page.

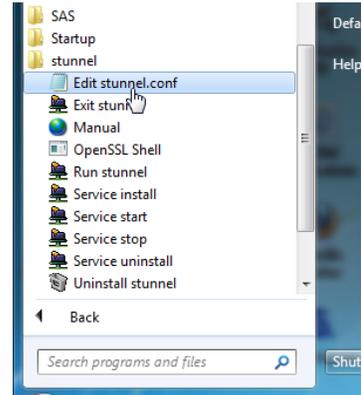
INSTALLING STUNNEL:

Download the most recent version of stunnel, and install. You will need Admin privileges to install.

<https://www.stunnel.org/index.html>

After installation, and while still using the Admin-privileged account, choose:

START->PROGRAMS->STUNNEL->Edit stunnel.conf



Replace the entire contents of stunnel.conf with the lines (green font) immediately below, and save:

```
# GLOBAL OPTIONS

client = yes
output = stunnel-log.txt
debug = 0
taskbar = no

# SERVICE-LEVEL OPTIONS

[SMTP Gmail]
accept = 127.0.0.1:1099
connect = smtp.gmail.com:465

[POP3 Gmail]
accept = 127.0.0.1:1109
connect = pop.gmail.com:995
```

Next choose: START->PROGRAMS->STUNNEL->Service install

Next choose: START->PROGRAMS->STUNNEL->Service start

Logoff the Admin-privileged account, reboot, and then login as your normal user.

Check to make sure that the stunnel service is automatically started (running) after the reboot, and available under the normal user account login:

Open the Windows "Task Manager": (CNTRL+SHIFT+ESC) or (CNTRL+ALT+DEL)

Select the Services Tab, and scroll down to the stunnel entry to make sure it is: 1) present, and 2) running.

