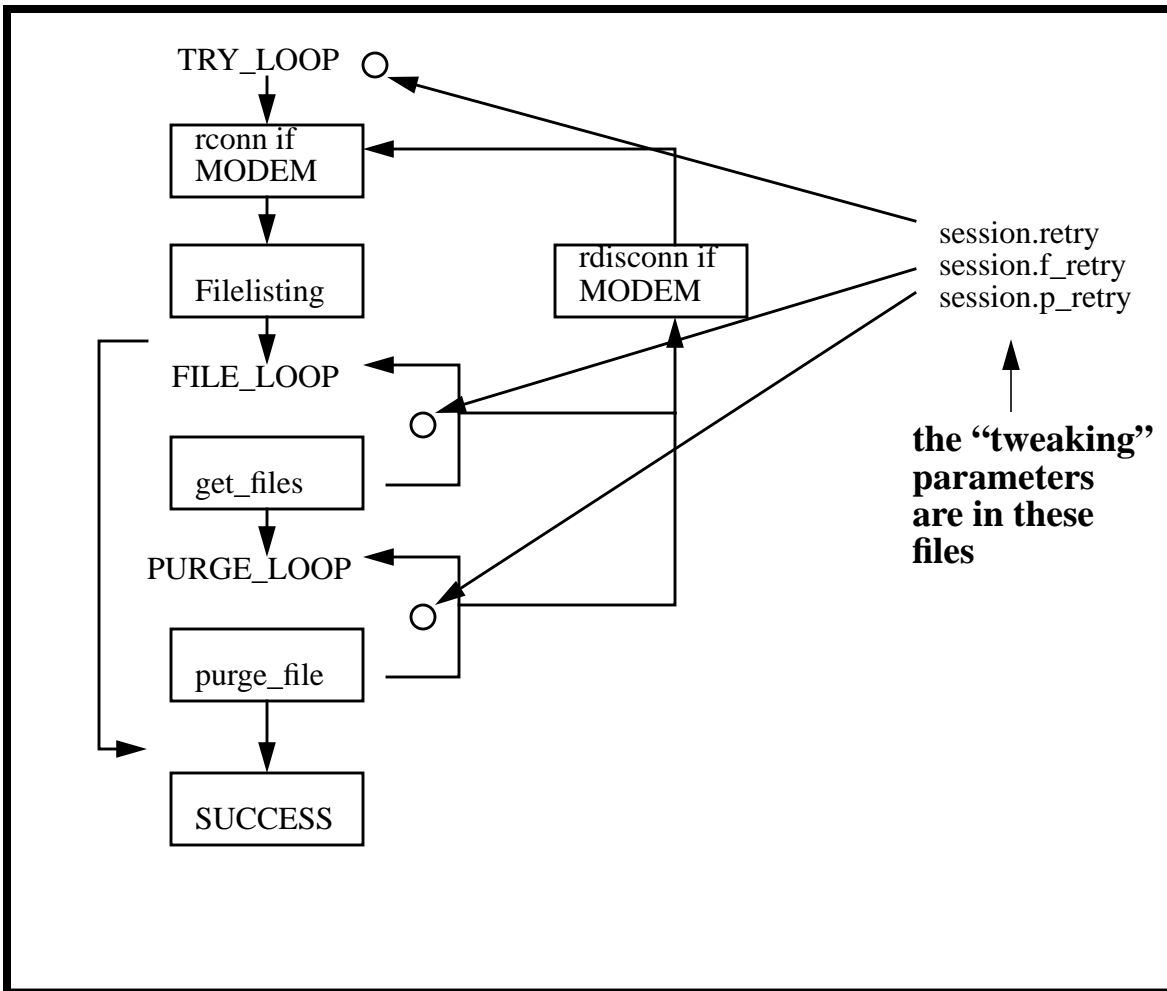


```
# Here is a sample SITE file called PUTZ.site for a DIRECT connection
# Comments can follow a "#" or any line without a ":" is ignored
# Relevant lines must be of the form "key : value"
# Whitespace is ignored
# Computer Specifics
env.rfilepath : bin/
env.downloadpath : Data/
env.modempath : Modem/
# Site specifics
site.id : PUTZ
site.name : This is a test site.
site.antenna_serial : 1234
site.computer_id : turbo
site.monument_id : mon4597
# Connection specifics
port.port : /dev/cua/c
# Valid choices are (1200,4800,9600,19200,38400)
port.speed : 19200
# Valid choices are (8N1,8N2,8E1,8O1,7E1,7O1)
port.control : 8N1
# Valid choices are (DIRECT, MODEM)
port.connection : DIRECT
# Modem specifics
modem.type : telebit
modem.attention : AT
modem.dialstring : ATDT4403304
modem.OK : OK
modem.reset : ATZ
modem.scale_factor : 3
# Session Specifics
session.elev_mask : 10
session.delta : 1
session.sync : 30
session.min_sv : 1
session.auto : YES
session.fastdownload : YES
session.retry : 3
session.f_retry : 3
session.p_retry : 3
```



An example **.site** file.

current logic, the receiver will continue to take data until the internal memory is full.

- `session.fastdownload` : *should the receiver use the fast download option* (Ex. YES)
This tells the scripts if they should use the rfile fast download option. Use this only in very clean connections (like direct cables). Valid values are “YES” and “NO”.
- `session.retry` : *number of times to attempt getting data* (Ex. 3)
This tells getall the number of times it will attempt to get data before it gives up. If you have a high error environment then increase this number. No files will be lost in a failure, but once the internal memory fills up the receiver will stop logging data.
- `session.f_retry` : *number of times to try and get a file after connected* (Ex. 3)
This number tells the script how many times to try and get data after connecting to the receiver. If you had a noisy line and were able to maintain the connection then you should make this number fairly large (10). After this number of tries is exhausted the script will hang up and try again the whole thing again for `session.retry` times.
- `session.p_retry` : *number of times to attempt a purge* (Ex. 3)
This number is similar to `session.f_retry` but tells the script how many times to attempt a purge. Unless you have very unusual circumstances this should probably stay at 3.

- `port.port` : *port or special file that receiver is connected to* (Ex. `/dev/cua/c`)
This is the name of the special file for the port that is connected to your receiver or communication device. You must have read/write permission for this file.
 - `port.speed` : *baud rate* (Ex. 19200)
This is the communication device interface speed (the speed between the computer and the first device attached to it). Valid choices are (1200,4800,9600,19200,38400).
 - `port.control` : *control characteristics bits, parity, stop bits* (Ex. 8N1)
This is the control characteristics that you want to use for the port. Valid choices are (8N1,8N2,8E1,8O1,7E1,7O1).
 - `port.connection` : *type of communication scheme used* (Ex. DIRECT)
This is type of connection. All connections can be characterized by whether you must send a “dialstring” (a Freewave radio-modem in Mode 6) or it acts like a direct connection (a Freewave radio-modem in a Master-Slave configuration). Valid choices are (DIRECT, MODEM).
 - `modem.type` : *name of modem configuration file* (Ex. telebit)
This is the full name of the configuration file that is used by rfile. This file should be located in the directory indicated by the `env.modempath` variable.
 - `modem.dialstring` : *modem command to dial remote site* (Ex. ATDT4403304)
This is the dial string sent to the modem. It must be a fully qualified modem command that starts with ATDT or ATDP. Commas represent pauses (length depends on your modem and its settings). Consult your modem manual for details and for possible case dependencies. This string will override the value in the configuration file.
 - `modem.attention` : *modem command to get attention* (Ex. AT)
 - `modem.config.1` : *string or command to send modem* (Ex. SomeString)
 - `modem.config.2` : *string or command to send modem* (Ex. SomeString)
 - `modem.config.3` : *string or command to send modem* (Ex. SomeString)
 - `modem.OK` : *modem return value for a successful command* (Ex. OK)
 - `modem.reset` : *command to reset modem* (Ex. ATZ)
 - `modem.hangup` : *command to hang up modem* (Ex. ATH or +++)
- These are not used currently. They will be used when Trimble releases the proprietary communication protocol. In that case the SERIAL.pm module is capable of dialing and making connections.
- `modem.scale_factor` : *scale factor used by rfile* (Ex. 3)
This is rfile specific multiplier that is used to determine the time-out constants used by rfile. If you have high latency in the line, try increasing this to 5 or 7. In general this should be 1.
 - `session.elev_mask` : *receiver elevation cutoff* (Ex. 10)
 - `session.sync` : *sample rate* (Ex. 30)
 - `session.min_sv` : *minimum of SV's to track* (Ex. 1)
 - `session.auto` : *should the receiver take this data every day* (Ex. YES)
- These are not used. Trimble_RFILE.pm and Trimble_RS232.pm are both capable of programming a receiver. It was determined that re-programming the receiver automatically places too much dependence on the computer being functional. If the computer fails in the

The .site file.

The **.site** file represents the permanent site to the software. **.site** files are stored in the directory called `~USER_PATH/Sites/`. Multiple **.site** files can be stored in the directory. All necessary site information is in the file and, assuming that you can see the Perl executable and the R-utility executables, lapdogs will download the data files from the receiver identified in the **.site** file. The name of the **.site** file is the argument that must be given to **dwncld_trimble**. It does not have to match the site.id variable (e.g. **dwncld_trimble** PUTZ). Note that **dwncld_trimble** expects only one site per call. (See `env.rfilepath` for possible problems in running multiple sites).

- `env.rfilepath` : *rfile executable path* (Ex. bin/)

This variable tells LAPDOGS where the R-Utility executables are located. Files must be visible to your operating account and have execute permission. This variable must end with a '/'. NOTE: Multiple runs of rfile should not be executed simultaneously from the same directory as the same configuration files will be overwritten by both. To simultaneously download two stations from the same computer, make a copy of your `~USER_PATH/bin` directory and modify this variable to reflect the path for the second station.

- `env.downloadpath` : *data download directory* (Ex. Data/)
- This is the location for the .r00 and .log files. It must end with a '/'.

- `env.modempath` : *directory in which modem configuration files are located* (Ex. Modem/)

This is the location of the modem configuration files used by the R-Utility programs. This variable ends with a '/'.

- `site.id` : *station ID* (Ex. PUTZ)

This is usually the four character station id. There is no restriction in the software for using a longer name. Allowable characters are restricted to those allowed in UNIX (no '/'s or '*'s). It is case dependent.

- `site.name` : *station description or long name* (Ex. This is a test site.)

This is a descriptive string and generally the long station name. It can be anything and any length, but it is suggested to keep it less than 256 characters for readability. Any longer and you may run into some unknown compiler restrictions.

- `site.antenna_serial` : *antenna serial number* (Ex. 000A39792)

This is the antenna serial number from the permanent stations. It is restricted to alphanumerics and underscores (in Perl speak [A-Za-z_0-9]).

- `site.computer_id` : *computer name or description* (Ex. zigzag.erl.mit.edu)

This is the name of the download computer where the software is running. You can use a short name like erde or a fully qualified name like erde.unavco.ucar.edu.

- `site.monument_id` : *UNAVCO assigned ID number* (Ex. mon4597)

This is a UNAVCO assigned number that uniquely describes the GPS monument. If you have not received this number from the UNAVCO archive please use the token "UNKNOWN".