Umbrella: A User-Interaction Language for Radio Astronomy at Haystack

John A. Ball and P. J. Charpentier
Haystack Observatory

Umbrella: A shade or screen especially for protection from rain or sun

Umbra: Complete shade or shadow; the darkest part of a shadow

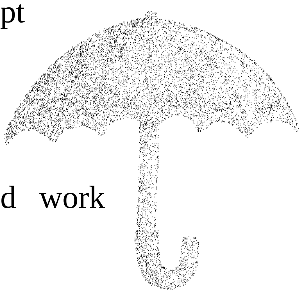
Contents

Introduction

Umbrella began as an interim user interface and scheduling system for radio-astronomical observing at Haystack. Beginning in 1994 October, Umbrella served as a user-interaction layer to isolate observers from the traditional Haystack operating system, now defunct. Combined with new pointing and data-acquisition systems, a new Umbrella now controls all single-antenna radio-astronomy operations at Haystack. As a result of recent improvements, manual operations, such as tuning local oscillators and changing cables and attenuators, are rarely necessary. For the few remaining such manual operations, Umbrella keeps track of what needs to be done and makes noises to wake users when assistance is needed.

Umbrella operates from one-word or one-line text commands, which can be typed by a user or, more typically, read from a previously prepared schedule.

To the extent possible, user interactions with the new Umbrella have been kept the same as before. With only minor changes, old Umbrella schedules and procedures should work with the new Umbrella.



Umbrella runs on gauss, a Linux machine, which also contains pointing programs, data acquisition programs, the VLBI Field System, data archives, and much else. The model that we had in mind for Umbrella is SNAP in the VLBI Field System. But procedures (alias macros) and schedule can be combined into a single file, if desired, and we have added looping and conditional constructs as illustrated below. Umbrella's language comprises primitives and "station" procedures, which are always available, and reads user-defined procedures and schedules from files that can be created using any editor capable of creating straight ASCII text—no hidden codes. Such files can be typed into an editor on gauss or fourier, read from floppy or Zip disks, or transferred over Internet.

Schedules

A schedule can call another schedule, in which case the original schedule is closed. That is, only one schedule is open at a time. A schedule can call itself, in which case it is restarted at the beginning or at a prescribed line number. This allows an additional looping construct. Schedules can execute any of the primitives and any previously defined procedures including those from previously read schedules. This allows a schedule to be separated into parts: one for procedures (macros) and another for execution. One schedule can be edited while another is in use. A procedure definition always replaces any other with the same name. The name of a procedure must not be the same as the name of a primitive.

Sources and Source Lists

Source lists are optional. The format is quite free and generally compatible with the old source-list format with the following changes:

1) Source names can be up to 40 characters long, but only the first 10 characters will be carried across to the header of the spectrum.

2) Embedded separators such as space tab = , (or) are not permitted in source names. Replace spaces in your source names, if any, with, for example, dots, underlines or hyphens.

3) Upper- and lower-case letters are equivalent, but whatever you type to Source = ... (rather than what is in the source list) will be carried and logged.

4) Names must be unique; Source = ... will find the first one that matches; others with the same name can never be found.

The format is:

SourceName RAhh RAmm RAss Decdd Decmm Decss Epoch Velocity

Velocity is the velocity in km/s referred to the local standard of rest (LSR) as used in spectroscopy.

Looping and Ifing

Waiting for or looping until a particular local sidereal time (LST) produces a conceptual problem: Which day? This would be a problem also for UT if the UT day were not specified. Sidereal day numbers are not in common use. We have chosen the following scheme to circumvent this problem: The wait-until command for LST, %, is to be followed by three numbers separated by colons, which will be interpreted as day number, hours, and minutes, just as for UT. The LST day number, however, is interpreted differently: 0 always means today (LST), 1 means tomorrow, and so forth. (0 for today is optional with UT.) The user needs to have an accurate notion of the sidereal time when the % command is to be executed. Sidereal wait-until commands to be executed near 0:00 LST are particularly tricky. If you specify %1:00:10, which means wait until tomorrow (LST) at 0 hours and 10 minutes, but the LST has already passed from 23 hours to 0 hours, then this command will wait almost a full day—probably not what you intended. Wait-until commands, ! and %, can be aborted by ^C or whatever keystroke you have set to be your interrupt. If the specified time has already passed, then either %... or !... produces no wait. These times depend on the accuracy of Umbrella's computer's clock, on the conversion from UT to LST, which is good to about a second, and on time quantization in Umbrella, which can be as much as two seconds.

There are now four possibilities for the arguments of DoLoops: A single integer means loop that many times. !*ddd*:*hh*:*mm* means loop until that UT day (day of the year) and time come up. Or a *ddd* of 0 means today, where today means the UT day when this line in the schedule is executed. +*hh*:*mm* means loop until *hh* hours and *mm* minutes from now, where now means the time when this line in the schedule is executed. And %*d*:*hh*:*mm* means loop until this local sidereal time (LST) comes up, where *d* = 0 means today, *d* = 1 means tomorrow, and so forth. Today means the LST (!) day when this line in the schedule is executed. The LST day increments at 00:00 LST.

The If construct takes a ! or % argument similar to DoLoop: If the time specified by !*ddd*:*hh*:*mm* or %*d*:*hh*:*mm* has *not* passed, then the commands inside the If (up to EndIf) are executed once; if the time has passed, then the commands inside the If are skipped.

We don't do seconds, but decimal minutes work. So, for example, +0:0.5 means for 30 seconds from now. All these time commands are precise at about ±2 seconds, but the time isn't checked again until all the commands inside a DoLoop are complete. Thus a long DoLoop may end much later than the prescribed time.

In order to guarantee that a DoLoop will be executed at least once, its end time must be later than the sum of the end time of the previous DoLoop or If plus the maximum time taken by whatever is inside the previous DoLoop or If. This is especially important at the end of a continuously running schedule where the last DoLoop typically ends at %0:24:00. This last DoLoop must be started before the end of the sidereal day lest Umbrella gets stuck therein.

We recommend Umbrella schedules that have all active commands inside DoLoops or Ifs with timed endings (! or %); only setup commands that apply to the whole schedule should be outside. Such a schedule can be started or restarted at any time because DoLoops and Ifs with times past will be skipped.

Because all (well, almost all) primitives check their present arguments against their last previous arguments and do nothing (well, almost nothing) if they are the same, there is almost no penalty for placing setup commands in DoLoops to be executed repeatedly.

DoLoops and Ifs cannot be nested; that is, only one DoLoop or If can be active at a time. This is a limitation that may be removed in a future version.

Primitives and Procedures

Primitives (functions) can be classified into three groups:

1) Action primitives produce some specified action such as moving the antenna or acquiring data. Example: DSS, which does a discrete source scan. Some action primitives take arguments, but, more typically, a primitive without an argument or with an optional argument is almost certainly an action primitive.

2) Conditional primitives may produce some action or not depending on the values of their arguments and the present state. Example: Source = ..., which slews the antenna to a new source unless its first argument, the source name, is the same as before, in which case it only notes this to the terminal.

3) Passive primitives produce a potential or actual change of state, depending on their arguments and the present state, but no action. Example: IntTime = ... records its argument in the shared-memory segment to be used later whenever an integration time is needed.

The order of commands in a schedule is often significant. As an example, to change to a source in a different source list, use SourceList = ... (a passive primitive) first, followed by Source = ..., (a conditional primitive), which slews the antenna to the new source. These two primitives might logically be put into a procedure named after the source.

Arguments are kept as character strings as you type them. Two arguments might have a different appearance but be functionally equivalent. Example: SetupRow = a and A are equivalent because case is ignored (except in file names).

Procedures typically contain primitives with arguments, and arguments on procedure command lines can also be passed to primitives therein using a $ notation as in Unix shell scripts. The first argument after a procedure is called $1, the second $2, and so forth. (Do not type these on the Define line.) Primitives within procedures can have $1, $2, and so forth as arguments, and these will be replaced by the corresponding arguments from procedure command lines. If nothing following a procedure command corresponds to a $ argument therein, then this $ argument will be replaced by a null. For most primitives, however, a null argument is different from no argument and is unlikely to produce useful results.

Procedures can contain other procedures (nested), which can contain other procedures, and so forth, to a depth that depends on the amount of stack memory available to the process. (In other words, we don't know how deep the nesting can go.) Arguments to procedures can be passed to procedures therein with the same $ scheme as for primitives.

For a listing of all currently defined primitives and procedures, type List. User-defined procedures will be at the end of this listing. To start a schedule, type Schedule = *filename*, where *filename* needs to include the Unix path if not the default. To interrupt a schedule, loop, or procedure in progress, type ^C or whatever your terminal uses for interrupt (intr). To quit, type Quit.

Primitives

Following is a list of primitives or built-in commands and a brief description of what each does. With any primitive that takes two or more arguments, you can type just the first argument to leave the others unchanged, or the first two arguments to leave those following unchanged, and so forth. With any primitive that needs an argument (not optional), and with most primitives that take arguments, typing just their name (no arguments) produces a note to the terminal with the present (or default) values of the arguments. Optional arguments are shown in brackets: [*arg*].

?

Same as Help.

Abort

Kill an observation in progress. This is sometimes needed because the spectrometer may continue even after you type ^C.

AzEl = *arg arg*

Slew the antenna to point to a fixed azimuth and elevation in degrees. The first *arg* is azimuth; the second *arg* is elevation. This is usually a test mode or to stow.

BandWidth = *arg*

Set the spectrometer's bandwidth to *arg* in MHz. *Important*: Select from this list only: 160, 53.3, 17.8, 5.93, 1.98, 0.66 MHz.

BeamSwitch = *arg*

If *arg* is On, spin up the beam switch, if *arg* is Off, spin it down and go to direct. Set Frequency first before BeamSwitch On. Default: Off.

Bell

Ring the terminal's bell and make other annoying noises to wake the operator.

Break

Break out of (terminate) a DoLoop. Note that a DoLoop with a break at the end is equivalent to an If. Commands after a break in a DoLoop cannot be executed. A Break in a schedule outside a DoLoop causes the *following* DoLoop or If to be skipped. Break typed by the operator has no effect; try ^C (that's <Control> C) instead. Break in Umbrella is not the same as break in C. See also Continue.

Cal

Do a noise calibration to measure system temperatures and, depending on CDPmode, also parameters to convert volts to antenna temperatures. Cal first checks and adjusts the power levels on source (to avoid saturation on strong sources) and then points the antenna to the off-source position prescribed by SwOff. (To calibrate on source, set SwOff = 0 0.) With cal noise on (vane in), Cal readjusts the levels downward if necessary, then measures power levels with cal noise on and off, and calculates system temperatures. The cal noise is left off and the antenna commanded back on source when Cal finishes. Redoing Cal is required after Frequency changes and recommended after any other significant changes such as Source, CDPmode, SetupRow, or Bandwidth.

CDPmode = *arg*

Set the mode of the continuum-data processor (CDP) to *arg*, where *arg* is a two-character string; the first character is a number, the second a letter (no space) as follows: Set the first character to 1 to use a single video converter (usually SetupRow = A), set it to 2 to use two video converters (usually SetupRow = C), and set it to 4 to use four video converters (usually SetupRow = E and hybrid mode). Set the second character of *arg* to U for unmodulated calibration noise (usually Beamswitch = Off), and set it to M for modulated cal (usually Beamswitch = On). Note that the U or M goes with the cal rather that with Dicke switching.

To get fast data recording, for example for Drift, set CDPmode = 1M; but you may, if need be, Cal with CDPmode = 1U (no beamswitching), then change to CDPmode = 1M for Drift or DriftP.

CompOffs = *arg*

If *arg* is Yes, then alternate offs in ON-OFFs will have their angular offsets complemented, that is, alternating in sign; if *arg* is No, then not complemented. Default: No.

Cont [= *arg*]

Same as Continue.

Continue [= *arg*]

Resume executing a schedule. With no *arg*, resume after the current line. With *arg* a positive integer, resume at line number *arg*, or with *arg* a negative integer, resume that many lines before the current line. *Important*: If the schedule is interrupted in a DoLoop or If (with ^C or an error), then Continue without an *arg* resumes with the line *after* that DoLoop or If. Continue = 1 will restart a schedule at the beginning. Continue in Umbrella is not the same as continue in C.

Contour [= *arg*]

Take data for a contour or other map. The mode is a boustrophedon scan in either azimuth and elevation or right ascension and declination as set by OffType. The scan rate is ScanRate, the width of the scan is twice HalfArc, the spacing between lines is Spacing, and the number of lines is NPoints, so the height of the scan is NPoints×Spacing. Data are recorded each IntTime seconds. If the optional *arg* is present, then it specifies the starting line number—default 1.

DickeS = *arg arg*

Set the Dicke switch frequency and blanking time. The first *arg* is Dicke switch frequency in Hz (default 10 Hz), and the second *arg* is blanking time as a percentage of each half cycle (default 20%). If in doubt, keep the defaults. There are restrictions on the values of these numbers: At this writing, the beamswitcher runs only at 10 Hz, and 20% blanking is needed. Frequency switching works fine with these defaults but can also switch faster, which might influence baseline shape, and with less blanking, which saves a little integration time. Both the Dicke half period and the blanking time must calculate to an integer number of milliseconds not less than 2. Check with our engineers before using non-default values. Changing DickeS clears Bandwidth; after DickeS, set or reset Bandwidth before observing.

DDS-DB

Do Double-Dicke-switched spectroscopy with double beam. DDS-DB puts the source alternately in the off-axis beam, as prescribed by OffBeam, and the on-axis beam while simultaneously beam switching. Each half of an off-on cycle lasts IntTime (in units of 30 seconds), and NPairs cycles combine into a spectrum.

Drift

Take data for a multiple-drift-curve map. The starting right-ascension and declination offsets are set by SwOff, the duration of each line is Duration, the declination step between lines is Spacing, and the number of lines is NPoints, so the height of the scan in declination is NPoints×Spacing. This whole cycle is repeated NPairs times. Data are recorded each IntTime seconds, and, if CDPmode is 1M and IntTime is 1, then raw data for each half cycle of DickeS (e.g., every 0.05 second for DickeS = 10 Hz) are also recorded to a file D.*yyyydddhhmm*, where *yyyy*, *ddd*, *hh*, and *mm* are the starting year, day, hour, and minute of UT. See also DriftP.

DriftP

Take data using multiple drift scans (MDS) for pointing measurements on weak sources in lieu of DSSs. First set IntTime = 1, CDPmode = 1M, and Npoints = 5 unless these are already set. Npairs sets the number of passes through the map; try Npairs = 1 for strong sources or more for weak sources. If your Frequency is less than 30 GHz (i.e., K band) then BeamSwitch and FrequencySwitch must both be off. Otherwise (i.e., Q band or 3 mm), either Beamswitch or FrequencySwitch must be on. DriftP calculates and sets Duration, Spacing, and SwOff and then does a Drift. More details are in a separate DriftP writeup.

DSS

Do continuum discrete-source scans for pointing. DSS does scans in azimuth and elevation each comprising about 50 data over a span of about five beamwidths centered, we hope, on the source. Each datum contains IntTime seconds of integration. The parameters of two Gaussians fitted to these scans are printed and logged to a file, S.*yyydddhhmm*, named as in Drift above. See also XferOffs.

Durat = *arg*

Same as Duration.

Duration = *arg*

Set duration time to *arg*, an integer in seconds. Duration is used in Drift. *Beware*: Duration is reset by DriftP. See also IntTime.

End

Same as Quit.

EX

Same as Quit.

Exit

Same as Quit.

FIVPT

Do a continuum five-point measurement for pointing. FIVPT takes six data, each with five IntTime seconds of integration, on an off position, a center position, and four positions offset from the center a half beamwidth in azimuth and elevation. The parameters of two Gaussians fitted to these data are printed and logged. FIVPT is still under construction. See also DSS.

Freq = *arg*

Same as Frequency.

FreqSOffs = *arg arg arg*

Set frequency offsets for frequency switching from these three *arg*s in kHz referred to the sky (not LORF). The first *arg* is signal, the second *arg* is comparison 1, and the third *arg* is comparison 2. The two comparison frequencies must be equal in magnitude but may, of course, differ in sign. FreqSOffs are used only when FrequencySwitch is On. To save time, set FreqSOffs first, then FrequencySwitch.

FreqSwitch = *arg*

Same as FrequencySwitch.

Frequency = *arg* [*arg*]

Set the line-rest frequency to the first *arg* in kHz. Frequency sets certain of the subreflector offsets and other parameters to default values for the receiver. The optional second *arg* is used in place of the calculated Foff for setting the OFFSET FREQUENCY dial on the LOs. Omit this second *arg* unless prescribed in the setup booklet for the receiver.

FrequencySwitch = *arg*

If *arg* is On, select frequency switching, if *arg* is Off, select no frequency switching. Default: Off. See also FreqSOffs.

HalfArc = *arg*

Set the half arc (i.e., half the width of the scan) to *arg* in degrees. Used in Contour.

Help

Print this help file to your screen through acroread. Use the features of acroread to, for example, find topics.

IFOff = *arg*

Same as IFOffset.

IFOffset = *arg*

Set the IF offset frequency to *arg* in kHz. This needs to be set only for special configurations: See the setup booklet for your receiver. *Important*: IFOffset is reset to receiver defaults after Frequency; that is, it is used just once.

IntTime = *arg*

Set integration time to *arg*, an integer. For continuum observations, the units are seconds; for spectroscopy, the units are 30 seconds, that is, IntTime = 2 means one minute for spectroscopy or two seconds for continuum. Default: 1.

List

Print all primitive and procedure names to your screen. Recently defined procedure names will be at the end of this listing.

ListS

List the sources in SourceList.

Maint = *arg*

Same as Maintenance.

Maintenance = *arg*

If *arg* is On, put the system into a peculiar maintenance mode, if *arg* is Off, put it back to normal. *Important*: For conventional observing, always set Maintenance = Off before setting other parameters such as Frequency.

More = *arg*

Give *arg*, the name of a text file, to the Unix function more, which allows the operator to read this file. Used for additional documentation. Type q to quit out of more back to Umbrella.

NPairs = *arg*

Number of pairs for continuum or spectral off-ons and for spectral antenna switching. Default: 1.

Npoints = *arg*

Number of points, IntTime each, for each off and each on for continuum off-ons or the number of switched runs (i.e., number of spectra) for spectral frequency switching (SFS) or beam switching (SBS). NPoints is also the number of lines in Contour and Drift. Default: 1.

OffBeam = *arg arg*

Set angular offsets to the off-axis beam or off-source position in azimuth (first *arg*) and elevation (second *arg*) in degrees replacing any previous offsets. This azimuth offset is cosine corrected to give arc on the sky. Default: −0.114 0.

OffOn

Same as ON-OFF.

OffType = *arg*

If *arg* is AzEl, then the off-source position in ON-OFFs will be offset in azimuth and elevation from OffBeam; if *arg* is RaDec, then in right ascension and declination from SwOff. OffType is also used to specify the coordinate system in Contour.

ON-OFF

Do continuum on-off measurements for source temperatures. For each on-off cycle, ON-OFF averages Npoints data, each with IntTime seconds of integration, at the source position and at a position offset in either azimuth and elevation or right ascension and declination: If OffType is AzEl, then the offset position in azimuth and elevation is prescribed by OffBeam. If OffType is RaDec, then the offset position in right ascension and declination is prescribed by SwOff. If CompOffs is Yes, then alternate offs will be complemented, that is, done with the opposite sign for RA or Az offset. ON-OFF repeats this cycle Npairs times and prints and logs the average differences with statistics. Under construction.

OnOff

Same as ON-OFF.

Park

Slew the antenna to the park-stow position: Az = 327° and El = 44°.75. At this position, you may shut off the drives, put in the stow pin, and leave; but no one may climb the antenna. See also Stow.

Pause

Wait for the operator to press Return. If the operator presses ^C and Return instead, then Umbrella will revert to command mode. See also Bell and Continue.

Project = *arg*

Set the project name or title to *arg*. Please set Project to your project's official name, for example, Salah-1.

Quit

I can't believe that you wouldn't want to use this program forever.

RADecOff = *arg arg arg*

Set right ascension and declination offsets in degrees. The first *arg* is RA offset, the second *arg* is Y (or yes) if you want this RA offset corrected for cosine of declination to give arc on the sky, or N (or no) if not. The third *arg* is declination offset. These offsets are limited to ±2°. *Important*: There is only one RA and one dec offset; RADecOff writes over whatever was there before. Several other functions use and write over these offsets. Example: ON-OFF. Default: 0 Y 0.

Reset

Reset almost all parameters to their default values. Reset does not remove or change any procedures that you have defined.

SAS

Do antenna-switched spectroscopy. SAS integrates IntTime (in units of 30 seconds) on each half of an off-on cycle and combines NPairs cycles into a spectrum. The offset to the off-source position is set by SwOff. CompOffs is *not* used. See also SOFFON.

SBS

Do beam-switched spectroscopy comprising Npoints spectra, each IntTime (in units of 30 seconds) long. See also DDS-DB.

ScanRate = *arg*

Set the scan rate to *arg* in degrees/second. Used in Contour.

Sched = *arg*

Same as Schedule.

Schedule = *arg* [*arg*]

Set the schedule-file name to the first *arg*. This file is to be on Umbrella's computer in the default directory unless this *arg* contains a path name. Unix file and path names are case sensitive. The ~ construct, as in csh or bash, also works. Umbrella starts executing this schedule at the first line or at the line number specified by the optional second *arg*. The Schedule command always closes and reopens the schedule file even if the name is the same. This ensures that a fresh (possibly edited) copy of the file is read from disk. Typing ^C or whatever you have set for interrupt will pause a schedule. See also Continue.

SetupRow = *arg*

Set the row in the spectrometer's table to *arg*. A means use one video converter, C means use two, and E means use all four video converters. Other rows in this table are not in use. Default: A.

SFS

Do frequency-switched spectroscopy to get Npoints spectra, each IntTime (in units of 30 seconds) long.

Sked = *arg*

Same as Schedule.

SOFFON

Do off-on (alias total-power) spectroscopy. SOFFON integrates IntTime (in units of 30 seconds) on each half of an off-on cycle and takes NPairs cycles to make NPairs spectra. The offset to the off-source position is prescribed by SwOff. CompOffs is *not* used. See also SAS.

SONOFF

Same as SOFFON.

Source = *arg* [*arg* ...]

With one *arg*, Source specifies either a planet or a source name to be looked for in SourceList. With more than one *arg*, they are taken to be in the form:

Source = *SourceName RAhh RAmm RAss DECdd DECmm DECss Epoch Velocity*

After the = sign, this is the same form as in SourceList. See also Velocity.

SourceList = *arg*

Set the source-list file name to *arg*. This file is to be on Umbrella's computer in the default directory unless *arg* contains a path name. Unix file and path names are case sensitive. The ~ construct, as in csh or bash, also works. The format of this file is discussed above. Source is reset to null by SourceList; a Source command should follow SourceList.

Spacing = *arg*

Set the spacing between lines to *arg* in degrees. Used in Contour and Drift. *Beware*: Spacing is reset by DriftP. Default: 0.

Stow

Slew the antenna to *near* the face-side-stow position: Az = 327° and El = 2°. The actual stow position is below this: El = −0°.25. At the actual stow position, you may shut off the drives, put in the stow pin, and an authorized person may climb the antenna. Please do *not* use Stow unless someone needs to climb the antenna; use Park instead.

SubTilts = *arg arg*

Set two subreflector tilts, azimuth and elevation in millidegrees. They are used to properly illuminate off-axis feeds and are normally set automatically. *Important*: These tilts are reset to receiver defaults by Frequency.

SubTrans = *arg arg arg*

Three subreflector translations in order: Z (focus), X, and Y in mils (milli-inches) are set by this command. Z (focus) is first because it is used more often. These translations can be used to maximize gain but are normally set automatically. *Important*: Z (focus) is reset to receiver defaults by Frequency.

SwOff = *arg arg*

Set angular offsets in right ascension and declination in degrees for the off-source position in antenna switching (SAS), off-ons (SOFFON or ON-OFF), or Cal, or the starting offsets for Drift. The first *arg* is in RA, the second *arg* is in Dec. These RA and Dec offsets to the off-source position are taken with respect to the original source position, that is, after removing the effect of any RADecOff. The SwOff RA offset is a coordinate offset; it is *not* cosine corrected to give arc on the sky. Normally the SwOff Dec offset should be zero, and this RA offset for spectroscopy should be minus IntTime times 0°.125 minus another 0°.05 or so for dead time moving the antenna. For spectroscopy with IntTime = 1, for example, try SwOff = −0.175 0, provided that there is no signal from this spot on the sky. SwOff offsets are limited to ±2°. Default: −0.175 0. *Beware*: SwOff is reset by DriftP. Do not confuse SwOff with RADecOff.

Test = *arg* [*arg* ...]

Echo all the *arg*s to your screen. This test can be useful if you are worried about separators in names.

UsrOffs = *arg arg*

Two user pointing offsets, azimuth and elevation in degrees, are set by this command replacing any previous offsets. Use DSSs or FIVPTs to check these. These offsets are limited to ±0°.1. Default: 0 0. See also XferOffs.

VCoffset = *arg arg arg arg*

Set up to four frequency offsets in MHz for the four video converters. VCoffsets are normally zero except for hybrid mode: See the setup booklet for the receiver. *Important*: The *arg*s after VCoffset must agree with the hardware; Umbrella has no way of checking the hardware setup. Default: 0 0 0 0.

Velocity = *arg*

Set the (LSR) Doppler velocity in km/s. This is the same parameter as the last *arg* after Source or in SourceList, so the Source command resets Velocity (to zero for planets). Use Velocity to change the velocity without changing sources or to offset from the center velocity of a planet.

XferOffs

Transfer the pointing offsets from the most recent DSS to UserOffs to affect subsequent pointing. XferOffs checks fairly carefully that this previous DSS produced good results; if not, XferOffs writes a warning message and does not change UserOffs.

There are three other symbols that act like primitives except that there must be no space following: ! % and +. These are wait-until commands discussed above under DoLoop, but they can also stand alone on a line, in which case Umbrella will wait while doing nothing.

Appendix A: Example Schedule

Following is an example of a user-created Umbrella schedule with comments. Spaces, tabs, =, and certain other symbols are separators and otherwise ignored. Each line is an entity. Anything after a # on each line is a comment. Blank lines are ignored. Upper- and lower-case letters are equivalent except in file names.

This example is a continuously running schedule, which means that it can be started anytime, and it will run forever, or at least a long time. By using sidereal times (% waits) and putting the essentials inside DoLoops and Ifs, the program will skip ahead in the schedule to the appropriate parts, regardless of when started, and recycle at the end (Continue = 1) when the sidereal day changes. A schedule like this but without planets could be started anytime in the year as well as anytime in the day; with planets, one needs to adjust the sources and times as the planets move around the sky.

```
# Example schedule for Umbrella at Haystack
# Revised:  1998 October 16, JAB

Maintenance = Off      # Always first
Project = Test-1       # Official name, please

Define  7-mmContinuum  # A setup procedure
  Frequency = 43122027 # kHz
  Bandwidth = 160      # For continuum, MHz
  BeamS                # Beam switching
                       # (from station procedure file)
  SetupRow = A         # One receiver (for DSSs and DriftP)
  CDPmode = 1M         # One receiver, modulated cal
EndDefine

Define  7-mmSpectroscopy
  Frequency = 43122027 # Line frequency, SiO masers, kHz
  Bandwidth = 17.8     # For spectroscopy, MHz
  BeamS                # Set beam switching
  SetupRow = C         # Two receivers (two polarizations)
  CDPmode = 2M         # Two receivers, modulated cal
EndDefine

Define  H2Ospectroscopy
  Frequency = 22235080 # Water masers, kHz
  Bandwidth = 17.8     # MHz
  FreqSOffs = -4444 4444 4444
                       # In-band offsets, kHz
  FreqS                # Frequency switching
  SetupRow = A         # One receiver
  CDPmode = 1U         # One receiver, unmodulated cal
EndDefine
```

```
Define  H2Ocontinuum   # For pointing on water masers
  Frequency = 22235080 # kHz
  Bandwidth = 0.66     # To match line widths, MHz
  # (We’ll use water maser features as pointing sources)
  FreqSOffs = 3000 -3000 3000
                       # Out-of-band offsets, kHz
  FreqS                # Frequency switching
  SetupRow = A         # One receiver
  CDPmode = 1U         # One receiver, unmodulated signal
EndDefine

Define  DoDSS          # Continuum discrete-source scan
  Cal                  # Calibrate
  IntTime = 1          # One second per point
  DSS                  # For pointing and gain
  XferOffS             # Use pointing offsets
EndDefine

Define  DoDriftP       # Drift map for pointing
  Cal                  # Calibrate
  IntTime = 1          # One second per point,
  CDPmode = 1M         # One receiver, and
  Npoints = 5          # Five rows are necessary for Drift
  Npairs = 2           # Two passes
  DriftP               # Drift-curve map
EndDefine

Define  DoDDS-DB       # Double-Dicke switching, dual beam
  BeamSwitch = On      # Beam switch should already be on
  Cal                  # Calibrate
  OffBeam = -0.114 0   # To off-axis beam
  IntTime = 2          # 60 seconds per half pair times
  Npairs = 6           # 6 pairs = 12 minutes per spectrum
  DDS-DB               # Spectroscopy
EndDefine

Define  DoSFS          # Frequency-switched spectroscopy
  Cal                  # Calibrate
  IntTime = 4          # Two minutes per spectrum
  Npoints = 1          # One spectrum
  SFS                  # Spectral frequency switched
EndDefine
```

```
# The action starts here:
DoLoop  %0:08:00       # Until 08:00 LST ...
  7-mmContinuum        # 43-GHz continuum
  Source = Saturn      # For pointing
  DoDSS                # Pointing check
  7-mmSpectroscopy     # For 43-GHz SiO masers
  #        Name    HH MM SS.S    DD MM SS   Epoch  Velocity
  Source = IKTau   03 50 43.6    11 15 32    1950  34.4
  DoDDS-DB             # Spectrum
  Source = TXCam   05 00 51.19   56 10 54    2000  11.0
  DoDDS-DB             # Spectrum
  Source = RCas    23 58 24.781  51 23 19.1  2000  25
  DoDDS-DB             # Spectrum
EndLoop                # Skipped if LST is past 08:00

DoLoop   %0:16:00      # Until 16:00 LST
  7-mmContinuum        # 43-GHz continuum
  Source = Venus       # For pointing and gain
  DoDSS                # Pointing checks
  Source = VirgoA  12 28 17.6   12 40 02  1950    0
  DoDriftP             # Drift map for gain
  7-mmSpectroscopy     # For 43-GHz SiO masers
  Source = SCrB    15 19 21.5   31 32 47  1950    0.5
  DoDDS-DB             # Spectrum
  Source = TCep    21 08 52.9   68 17 12  1950   -7.3
  DoDDS-DB             # Spectrum
EndLoop                # Skipped if LST is past 16:00

DoLoop  %0:24:00       # The last DoLoop of a recycling
  H2OContinuum         # schedule should end at 24:00
  Source = W3OH    02 23 17.8   61 38 58  1950  -48
  DoDSS                # Peak up on maser feature
  H2OSpectroscopy      # Switch to spectroscopy
  DoSFS                # Frequency-switched spectroscopy
  H2OContinuum         # Back to 13-mm continuum
  Source = W75N    20 36 50.5   42 27 01  1950   13
  DoDSS                # Peak up on maser feature
  H2OSpectroscopy      # Switch to spectroscopy
  DoSFS                # Frequency-switched spectroscopy
EndLoop

# Everybody should recycle:
Continue = 1           # Restart the same schedule
```

Appendix B: The Station Schedule

```
# This is the station schedule file for Umbrella.
# Revised:  1998 October 12, JAB
Define  ReadAll        # Read all (well, almost all) the params
  AzEl
  BandWidth
  BeamSwitch
  CDPmode
  CompOffs
  DickeS
  Duration
  FreqSOffs
  Frequency
  FrequencySwitch
  HalfArc
  IFOffset
  IntTime
  Maintenance
  NPairs
  NPoints
  OffBeam
  OffType
  Project
  RADecoff
  ScanRate
  Schedule
  SetupRow
  Source
  SourceList
  Spacing
  SubTilts
  SubTrans
  SwOff
  UsrOffs
  VCoffset
  Velocity
EndDefine

Define  FreqS          # Change to frequency switching
  BeamSwitch = Off
  DickeS = 10, 10      # Hz, %
  FreqSOffS            # List for reference
  FrequencySwitch = On
EndDefine
```

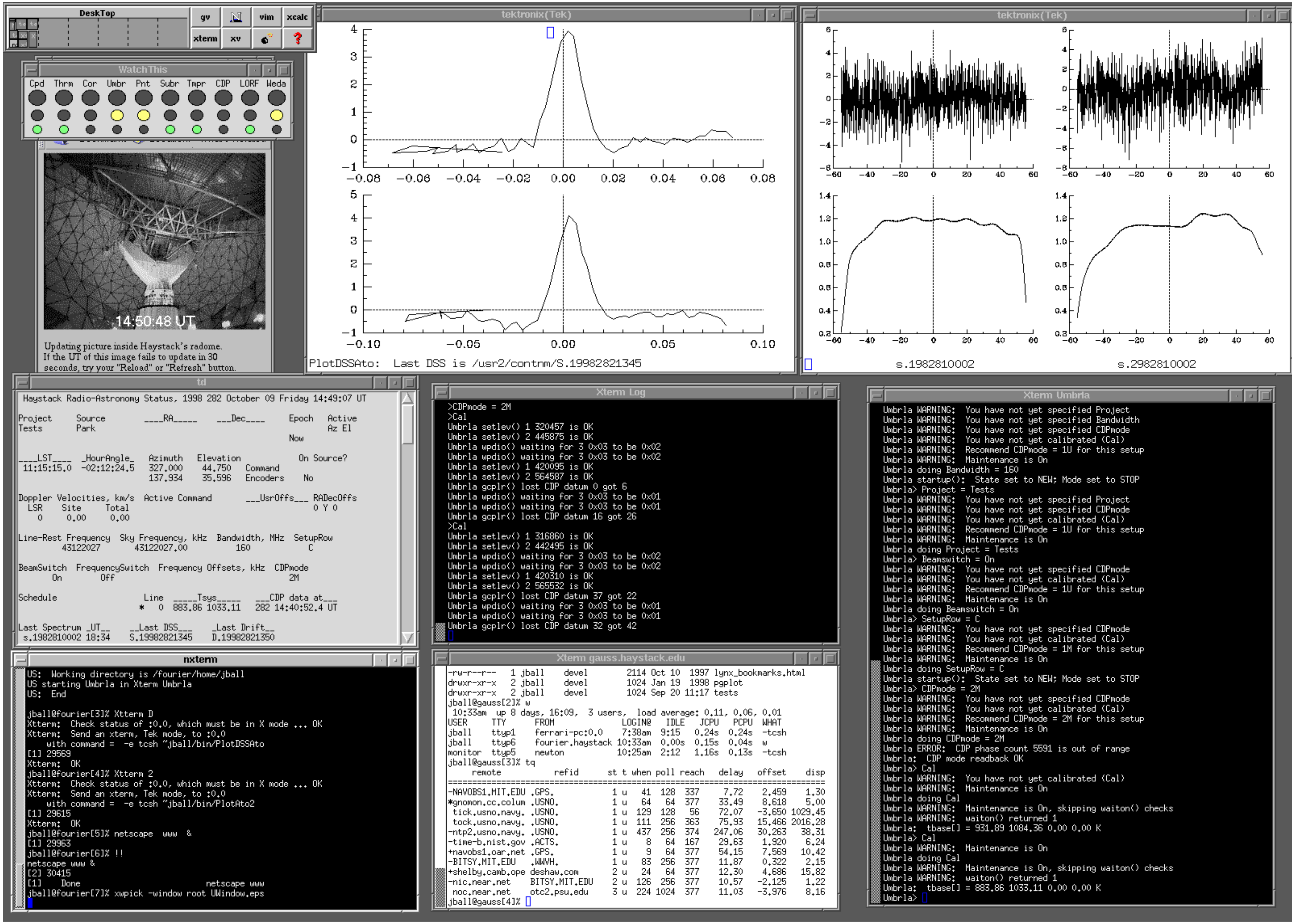
```
Define  BeamS          # Change to beam switching
  FrequencySwitch = Off
  DickeS = 10, 24      # Hz, %
  BeamSwitch = On
EndDefine

Define  CMVA           # Setup for CMVA VLBI with Field System
  Maintenance = Off
  Project = CMVA
  Frequency = 86040000 # kHz
  Bandwidth = 160      # MHz
  SetupRow = C         # Two video converters
  BeamSwitch = Off
  FrequencySwitch = Off
  CDPmode = 2U
  Park                 # Until a source from the Field System
EndDefine
```

Appendix C: Picture of Umbrella Running

Below is a screen snapshot of fourier's console showing Umbrella running on gauss. At the lower right, labeled Xterm Umbrla, is the Umbrella command window. In the center, labeled Xterm Log, is the logging window used by UPoint, Dop, and some logging also from Umbrella. Near the upper left is WatchThis. These three windows were started by the US script. (Well, we moved them around.) At center left, labeled td, is a text display from clicking on Umbr in WatchThis. (That's t for text, and the traffic lights are lettered from a to j, left to right; so Umbr's text display is td.) Just below WatchThis is an updating picture taken inside Haystack's radome as displayed by Netscape: http://yoda.haystack.edu/Ant.html. At top center is a quick plot of the last DSS from Xtterm D, and at top right, a quick plot of the last two (simultaneous) spectra from Xtterm 2. At bottom left and bottom center are two general-purpose xterms with shells from fourier and gauss. Everything else was started from this nxterm on fourier.

Revised: 1998 October 17, JAB



Updating picture inside Haystack's radome.
If the UT of this image fails to update in 30
seconds, try your "Reload" or "Refresh" button.

```
PlotDSSAto:  Last DSS is /usr2/contnm/S.19982821345
```

```
 Haystack Radio-Astronomy Status, 1998 282 October 09 Friday 14:49:07 UT

Project     Source        ____RA_____    ___Dec____     Epoch   Active
Tests       Park                                                Az El
                                                        Now

____LST____  _HourAngle_   Azimuth   Elevation              On Source?
 11:15:15.0  -02:12:24.5   327.000    44.750   Command
                           137.934    35.596   Encoders      No

Doppler Velocities, km/s  Active Command        ___UsrOffs___ RADecOffs
  LSR     Site     Total                                       0 Y 0
   0      0.00     0.00

Line-Rest Frequency  Sky Frequency, kHz  Bandwidth, MHz  SetupRow
         43122027       43122027.00           160            C

BeamSwitch  FrequencySwitch  Frequency Offsets, kHz  CDPmode
       On        Off                                    2M

Schedule                     Line  _____Tsys_____   ___CDP data at___
                           *   0  883.86 1033.11   282 14:40:52.4 UT

Last Spectrum _UT__    __Last DSS___    _Last Drift__
 s.1982810002 18:34    S.19982821345    D.19982821350
```

```
>CDPmode = 2M
>Cal
Umbrla setlev() 1 320457 is OK
Umbrla setlev() 2 445875 is OK
Umbrla wpdio() waiting for 3 0x03 to be 0x02
Umbrla wpdio() waiting for 3 0x03 to be 0x02
Umbrla setlev() 1 420095 is OK
Umbrla setlev() 2 564587 is OK
Umbrla gcplr() lost CDP datum 0 got 6
Umbrla wpdio() waiting for 3 0x03 to be 0x01
Umbrla wpdio() waiting for 3 0x03 to be 0x01
Umbrla gcplr() lost CDP datum 16 got 26
>Cal
Umbrla setlev() 1 316860 is OK
Umbrla setlev() 2 442495 is OK
Umbrla wpdio() waiting for 3 0x03 to be 0x02
Umbrla wpdio() waiting for 3 0x03 to be 0x02
Umbrla setlev() 1 420310 is OK
Umbrla setlev() 2 565532 is OK
Umbrla gcplr() lost CDP datum 37 got 22
Umbrla wpdio() waiting for 3 0x03 to be 0x01
Umbrla wpdio() waiting for 3 0x03 to be 0x01
Umbrla gcplr() lost CDP datum 32 got 42
```

```
Umbrla WARNING:  You have not yet specified Project
Umbrla WARNING:  You have not yet specified Bandwidth
Umbrla WARNING:  You have not yet specified CDPmode
Umbrla WARNING:  You have not yet calibrated (Cal)
Umbrla WARNING:  Recommend CDPmode = 1U for this setup
Umbrla WARNING:  Maintenance is On
Umbrla doing Bandwidth = 160
Umbrla startup():  State set to NEW; Mode set to STOP
Umbrla> Project = Tests
Umbrla WARNING:  You have not yet specified Project
Umbrla WARNING:  You have not yet specified CDPmode
Umbrla WARNING:  You have not yet calibrated (Cal)
Umbrla WARNING:  Recommend CDPmode = 1U for this setup
Umbrla WARNING:  Maintenance is On
Umbrla doing Project = Tests
Umbrla> Beamswitch = On
Umbrla WARNING:  You have not yet specified CDPmode
Umbrla WARNING:  You have not yet calibrated (Cal)
Umbrla WARNING:  Recommend CDPmode = 1U for this setup
Umbrla WARNING:  Maintenance is On
Umbrla doing Beamswitch = On
Umbrla> SetupRow = C
Umbrla WARNING:  You have not yet specified CDPmode
Umbrla WARNING:  You have not yet calibrated (Cal)
Umbrla WARNING:  Recommend CDPmode = 1M for this setup
Umbrla WARNING:  Maintenance is On
Umbrla doing SetupRow = C
Umbrla startup():  State set to NEW; Mode set to STOP
Umbrla> CDPmode = 2M
Umbrla WARNING:  You have not yet specified CDPmode
Umbrla WARNING:  You have not yet calibrated (Cal)
Umbrla WARNING:  Recommend CDPmode = 2M for this setup
Umbrla WARNING:  Maintenance is On
Umbrla doing CDPmode = 2M
Umbrla ERROR:  CDP phase count 5591 is out of range
Umbrla:  CDP mode readback OK
Umbrla> Cal
Umbrla WARNING:  You have not yet calibrated (Cal)
Umbrla WARNING:  Maintenance is On
Umbrla doing Cal
Umbrla WARNING:  Maintenance is On, skipping waiton() checks
Umbrla WARNING:  waiton() returned 1
Umbrla:  tbase[] = 931.89 1084.36 0.00 0.00 K
Umbrla> Cal
Umbrla WARNING:  Maintenance is On
Umbrla doing Cal
Umbrla WARNING:  Maintenance is On, skipping waiton() checks
Umbrla WARNING:  waiton() returned 1
Umbrla:  tbase[] = 883.86 1033.11 0.00 0.00 K
Umbrla>
```

```
US:  Working directory is /fourier/home/jball
US starting Umbrla in Xterm Umbrla
US:  End

jball@fourier[3]% Xtterm D
Xtterm:  Check status of :0.0, which must be in X mode ... OK
Xtterm:  Send an xterm, Tek mode, to :0.0
    with command =  -e tcsh ~jball/bin/PlotDSSAto
[1] 29569
Xtterm:  OK
jball@fourier[4]% Xtterm 2
Xtterm:  Check status of :0.0, which must be in X mode ... OK
Xtterm:  Send an xterm, Tek mode, to :0.0
    with command =  -e tcsh ~jball/bin/PlotAto2
[1] 29615
Xtterm:  OK
jball@fourier[5]% netscape  www  &
[1] 29963
jball@fourier[6]% !!
netscape www &
[2] 30415
[1]    Done                          netscape www
jball@fourier[7]% xwpick -window root UWindow.eps
```

```
-rw-r--r--   1 jball    devel        2114 Oct 10  1997 lynx_bookmarks.html
drwxr-xr-x   2 jball    devel        1024 Jan 19  1998 pgplot
drwxr-xr-x   2 jball    devel        1024 Sep 20 11:17 tests
jball@gauss[2]% w
 10:33am  up 8 days, 16:09,  3 users,  load average: 0.11, 0.06, 0.01
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU  WHAT
jball    ttyp1    ferrari-pc:0.0    7:38am  9:15   0.24s  0.24s  -tcsh
jball    ttyp6    fourier.haystack 10:33am  0.00s  0.15s  0.04s  w
monitor  ttyp5    newton           10:25am  2:12   1.16s  0.13s  -tcsh
jball@gauss[3]% tq
     remote           refid      st t when poll reach   delay   offset    disp
==============================================================================
-NAVOBS1.MIT.EDU .GPS.            1 u   41  128  337     7.72    2.459    1.30
*gnomon.cc.colum .USNO.           1 u   64   64  377    33.49    8.618    5.00
 tick.usno.navy. .USNO.           1 u  129  128   56    72.07   -3.650 1029.45
 tock.usno.navy. .USNO.           1 u  111  256  363    75.93   15.466 2016.28
-ntp2.usno.navy. .USNO.           1 u  437  256  374   247.06   30.263   38.31
-time-b.nist.gov .ACTS.           1 u    8   64  167    29.63    1.920    6.24
+navobs1.oar.net .GPS.            1 u    9   64  377    54.15    7.569   10.42
-BITSY.MIT.EDU   .WWVH.           1 u   83  256  377    11.87    0.322    2.15
+shelby.camb.ope deshaw.com       2 u   24   64  377    12.30    4.686   15.82
-nic.near.net    BITSY.MIT.EDU    2 u  126  256  377    10.57   -2.125    1.22
 noc.near.net    otc2.psu.edu     3 u  224 1024  377    11.03   -3.976    8.16
jball@gauss[4]%
```