LWA Station-Level Observing Procedure and Associated Metadata

Version 10

S. Ellingson, J. Dowell

February 2026

Long Wavelength Array Project

Change Record

Version	Date	Author(s)	Changes
1	2010-06-16	S. Ellingson	• First version.
2	2011-02-27	S. Ellingson	• Renamed `tp_session_sch` to `tpss`; UT changed to UTC; mean Julian date changed to modified Julian date; `OBS_START_MPM` range clarified; `OBS_RA`/`OBS_DEC` epoch J2000; SDF lines required order; `OBS_BW`=5 now 4.9 MSPS; `PROJECT_ID` 8 chars; session-specific optional params reordered; Added `SESSION_CRA`; TBN freq set by `OBS_FREQ1`; Session Spec File format significantly different.
3	2011-03-01	S. Ellingson	• Replaced SDF example in Appendix 1.
4	2011-03-29	S. Ellingson	• Added `DIAG1` mode; `OBS_STP_T[`n`]` valid values updated; Deleted `OBS_COMMENT` codes; `SESSION_DRX_BEAM` now 2-byte signed; `SESSION_LOG_`/`SESSION_INC_` now 1-byte signed; In obs spec, `OBS_START_MJD`/`MPM`/`DUR` now 8-byte unsigned; Changed session metadata file; Additional files in tarball; Clarified dynamic subdirectory; Fixed minor typos.
5	2011-04-12	S. Ellingson	• Added tag field to session metadata; Explained both `OBS_FEE` must be off; Fixed minor typos.

Version	Date	Author(s)	Changes
10	2026-02	J. Dowell	• All DP_ references changed to NDP throughout. • Removed TBW and TBN observing modes; added TRK_LUN, TBT, TBS. • Added SESSION_SPC, OBS_BDM keywords. • OBS_ASP_ATS replaced by OBS_ASP_AT3 (range 0–31). • Removed OBS_TBW_BITS, OBS_TBW_SAMPLES, OBS_TBN_GAIN. • Added OBS_TBT_SAMPLES. • Stand counts updated from 260 to 256; beam delays from 520 to 512. • OBS_DRX_GAIN range updated to 0–255 (packed per-tuning gains). • OBS_ASP_FLT range updated to 0–7. • Binary formats updated: SESSION_SPC, OBS_BDM fields added; SESSION_MRP_NDP/MUP_NDP replace DP_. • Updated session output tarball contents to match current implementation. • Updated session metadata file format (added BARCODE field, documented outcome codes). • Added implementation notes for 256-stand array sizing and stepped observation setting carry-over. • Removed OBS_BW sample rate table; refer to NDP ICD. • Updated organization address to University of New Mexico.

Contents

1 Introduction

This memo documents the procedure for using the monitoring and control system (MCS) to conduct observations with an LWA station. This includes the procedure for designing observations, as well as the process for working with MCS to schedule and conduct the observation. This memo also serves to document the content and format of observation “metadata”; that is, data which is produced by MCS as part of the processes of scheduling and conducting the observation. Metadata is distinct from the primary instrument output, which is captured by MCS data recorders (MCS-DRs). The latter is documented in the ICDs of the NDP and MCS-DR subsystems ([1] and [2], respectively).

2 Definitions

A *project* is a collection of activities conducted for a common scientific purpose. Every project has a single principal investigator (PI). A project consists of one or more *sessions*. A *session* is a contiguous block of time during which MCS is conducting observations on behalf of a single project, using a single beam. A session consists of one or more *observations*. An *observation* is a single scan using a single mode (defined below). The following modes of observation are defined:

`TRK_RADEC` – RA/DEC Tracking. A beam is formed in a specified direction (right ascension and declination) and follows that direction on the sky.

`TRK_SOL` – Solar Tracking. A beam is formed in the direction of the Sun and follows the Sun across the sky.

`TRK_JOV` – Jupiter Tracking. A beam is formed in the direction of Jupiter and follows Jupiter across the sky.

`TRK_LUN` – Lunar Tracking. A beam is formed in the direction of the Moon and follows the Moon across the sky.

`STEPPED` – Stepped Tracking. A beam is pointed in a sequence of specified directions, with specified dwell times. Pointing directions can be specified either in equatorial (RA/DEC) or horizontal (azimuth/altitude) coordinates. The beam pointing is fixed in the specified coordinate system until the next repointing. One may also fix the pointing for the duration of the entire observation. In addition, the center frequency of each tuning may be changed at each step.

`TBT` – Transient Buffer – Triggered. A dump of the transient buffer of the digital processor (NDP). The duration is computed automatically from the requested number of samples.

`TBS` – Transient Buffer – Streaming. A continuous capture from the transient buffer of NDP. The duration must be specified by the PI.

`DIAG1` – Diagnostic Mode 1. In this mode, all parameters other than `OBS_MODE` are ignored. MCS returns the metadata file immediately. This mode is useful for verifying that MCS and the Executive are operational.

As an example, consider a project consisting of 10 sessions, where each session consists of 3 observations. During each session, MCS first forms a beam on a calibrator for 5 minutes (`TRK_RADEC`), then tracks a target for 1 hour (`TRK_RADEC`), and finally tracks the same calibrator again for 5 minutes (`TRK_RADEC`). All three observations in each session use the same beam, tunings, and bandwidth. Each session takes place on a different day.

3 Process

The process for defining, scheduling, and conducting observations is as follows:

1. The PI creates a session definition file (SDF). This is a human-readable text file whose format is described in Section 4. Each SDF defines exactly one session.
2. An operator submits the SDF using the task processor program `tpss`. The following occurs:
   (a) The SDF is parsed and checked for errors, and any derived information is displayed to the operator for confirmation.
   (b) Resources are assigned. For example, it is at this point that the operator might assign a particular NDP beam output to the session. Concurrently, `tpss` is checking for resource conflicts (e.g., making sure the assigned beam is not already allocated at the requested time). The operator also ensures that a reasonable value is assigned to `SESSION_CRA` (see description below), and that this value is consistent with other concurrently-running sessions and the station-level CRA policy.
   (c) The operator edits the SDF `REMPO` fields (`PROJECT_REMPO` and `SESSION_REMPO`) as appropriate. These are remarks pertaining to review and scheduling that the operator wishes to convey as metadata. `tpss` is run at least once so that these comments are included in the version of the SDF that is submitted to MCS/Executive.
   (d) `tpss` outputs the following files:[1]
      - The SDF, now modified as a result of the steps above. In this version of the SDF, the value assigned to every keyword is explicitly indicated. The name of this file has the format `${PROJECT_ID}_${SESSION_ID}.txt`.
      - A *session specification file*, named `${PROJECT_ID}_${SESSION_ID}.ses`. This is a binary file that specifies parameters that apply session-wide. These parameters are defined in Section 5.
      - One *observation specification file* per observation, named `${PROJECT_ID}_${SESSION_ID}_${OBS_ID}.obs`. These are binary files that completely specify the scheduled observations. The format of these files is given in Section 6. It is these files, plus the session specification file, that are the actual input to MCS/Executive for conducting observations.
3. The session runs at the scheduled time. Primary NDP output data is captured by MCS-DR.
4. As each observation concludes, MCS creates a modified copy of the observation specification file with filename `${PROJECT_ID}_${SESSION_ID}_${OBS_ID}_${OBS_OUTCOME}.dat`. `${OBS_OUTCOME}` is an integer which indicates the outcome of the observation, such as whether the observation succeeded or failed (see Section 7). For parameters in the SDF which were set to "MCS Decides", the contents of this file reflect the actual values used.
5. At the conclusion of the session, MCS creates a gzipped tarball named `${PROJECT_ID}_${SESSION_ID}.tgz` containing the following:
   - The SDF, `.ses` file, and `.obs` files.

[1] In the case of `STEPPED` mode observations, the keywords associated with step definitions are not written into the modified SDF but instead are written directly into the observation specification file.

- An .ipl file ("inprocessing log").
- A .cs file ("command script").
- A session metadata file, named ${PROJECT_ID}_${SESSION_ID}_metadata.txt (see Section 7).
- An mcs.host file containing the hostname of the MCS computer. This is useful for determining which station produced a given metadata file.
- A mindelay.txt file containing minimum beam delay information. This is useful for correlating data from multiple stations when using the LWA Swarm as an interferometer (see [6]).
- Optionally, the station static MIB (ssmif.dat), if SESSION_INC_SMIB was set. The format of this file is described in [3].
- ASP MIB snapshots taken at the beginning and end of each observation (ASP_begin.gdb and ASP_end.gdb).
- A "dynamic" subdirectory containing sdm.dat (Station Dynamic MIB). The format of this file is described in [3].
- If SESSION_LOG_SCH= 1, the MCS/Scheduler log file, mselog.txt, is included; it is edited to cover the time period of the session.
- If SESSION_LOG_EXE= 1, the MCS/Executive log file, meelog.txt, is included in the "dynamic" subdirectory; it is edited to cover the time period of the session.

4 Format of a Session Definition File

Session definition files (SDFs) are human-readable text files. See the example provided in Appendix A.

Session definition files consist of lines, with each line having the following structure:

- A keyword identifying a parameter
- At least one whitespace character
- Data intended to be assigned to the parameter
- Newline (line terminator)

A line may be up to 4096 characters long. Lines which are empty (i.e., containing only the newline character) are allowed, ignored, and encouraged as a way to improve the readability. The "data" field contains only alphanumeric characters (including space) plus standard punctuation and common symbols, but not special/invisible characters. Note that any whitespace between the end of the intended data and the newline is significant.

Generally, session definition files have three or more parts. The first part is a set of lines identifying the PI and project. The second part is a set of lines identifying the session, including parameters that apply session-wide. The third part is a set of lines identifying the first observation, including parameters that apply to that observation. Each additional observation is defined by repeating the third part with the desired modifications. Only parameters which are different from the previous observation need to be defined for subsequent observations.

The following is a list of defined parameters, in the order in which it is required that they appear in the session definition file. In each case, we give the identifying keyword, followed by information on valid values.

4.1 PI/Project Identification

The following keywords define the PI and project:

`PI_ID` – PI Identification. This is intended to enable concise, unambiguous identification of the PI. PI identification codes should be assigned and maintained by the Project Office. It is recommended that this be a sequentially-assigned integer.

`PI_NAME` – PI Name. This is redundant information given `PI_ID`, but is included primarily for user convenience. The recommended format is *Last_Name, First_Name Middle_Initial(s)*.

`PROJECT_ID` – Project Identification. This is a string of no more than 8 characters. This is intended to enable concise, unambiguous identification of the project. Project identification codes should be assigned and maintained by the Project Office. Since this is used as part of the filename of some files, it is strongly recommended that these be minimum length, free of whitespace, and constructed to be easy to sort; for example "E00037" where "E" identifies a class of projects and "00037" means the 37th project in this class.

`PROJECT_TITLE` – This is redundant information given `PROJECT_ID`, but is included primarily for user convenience.

`PROJECT_REMPI` – Remarks from the PI on the project. This is intended to convey information that might not be present or obvious through other session definition file parameters.

`PROJECT_REMPO` – Remarks from the Project Office on the project. This is intended to convey information that might not be present or obvious through other session definition file parameters.

4.2 Session Identification

The following keywords identify the session and provide session-wide parameters:

`SESSION_ID` – Session Identification. This is intended to enable concise, unambiguous (in combination with `PROJECT_ID`) identification of the session. Session identification codes are sequentially-assigned integers, beginning with 1.

`SESSION_TITLE` – This is redundant information given `SESSION_ID`, but is included primarily for user convenience.

`SESSION_REMPI` – Remarks from the PI on the session. This is intended to convey information that might not be present or obvious through other session definition file parameters.

`SESSION_REMPO` – Remarks from the Project Office on the session. This is intended to convey information that might not be present or obvious through other session definition file parameters.

4.2.1 Optional Session Parameters

The following session-level parameters are optional. If not specified, MCS assigns default values.

`SESSION_CRA` – Configuration Request Authority. An unsigned integer in the range 0–65535 which indicates the priority of this session (with respect to other concurrent sessions) in determining authority to set FEE and ASP settings, which affect all concurrent sessions. "0" (the default if not specified) means no authority; the keywords `OBS_FEE` and `OBS_ASP_*` will simply be ignored. When multiple sessions are running concurrently on the station and at least one of them has `SESSION_CRA`>0, the session with the largest `SESSION_CRA` sets FEE and ASP settings within the constraints of the station CRA policy set in the SSMIF. In the event of a tie, MCS will proceed as if all concurrent sessions have `SESSION_CRA`$=0$.

`SESSION_DRX_BEAM` – DRX beam to use for this session. An integer in the range 1–4. A value of -1 means MCS decides (value used if not specified). This is not meaningful if `OBS_MODE` = `TBT` or `TBS`, and is ignored for those observing modes.

`SESSION_SPC` – Spectrometer configuration for the MCS data recorder (MCS-DR). A free-form string of up to 31 characters. This string is passed directly to the DR as the argument of the `SPC` command; see [2] for details on the format and valid values. If not specified, the spectrometer is not used.

`SESSION_MRP_`*`sss`* – MIB recording period for subsystem *sss*, where *sss* is one of `ASP`, `NDP`, `DR1`, `DR2`, `DR3`, `DR4`, `DR5`, `SHL`, or `MCS`. Integer minutes. For example, `SESSION_MRP_ASP` = 5 will cause MCS to archive a copy of the ASP MIB every 5 minutes for the duration of the observation. "0" = "never record", and "-1" = "MCS decides" (value used if not specified). Note that the setting of this parameter does not imply anything about how often the MIB is *updated*; see `SESSION_MUP_`*`sss`*. Typically, users will want `SESSION_MRP_`*`sss`* $\geq$ `SESSION_MUP_`*`sss`*. When invoked, the order of invocation of subsystems must be: ASP, NDP, DR1, DR2, DR3, DR4, DR5, SHL, MCS.

`SESSION_MUP_`*`sss`* – MIB update period for subsystem *sss*, where *sss* is as defined above. Integer minutes. For example, `SESSION_MUP_ASP` = 5 will request MCS to force a 100% update of the ASP MIB every 5 minutes for the duration of the observation. "0" = "request no updates (but don't prevent them either)", and "-1" = "MCS decides" (value used if not specified). It should be noted that there is only one set of MIBs for the station, and that they are common to all sessions. Therefore, if MCS or some other concurrently-running session successfully requests a shorter update period, then this parameter will have no effect. When invoked, the order of invocation of subsystems must be: ASP, NDP, DR1, DR2, DR3, DR4, DR5, SHL, MCS.

`SESSION_LOG_SCH` – If this is "1", the portion of the MCS/Scheduler log file (`mselog.txt`) corresponding to the time period of the session is saved as metadata. "0" = "don't save", which is assumed if the parameter is not specified.

`SESSION_LOG_EXE` – If this is "1", the portion of the MCS/Executive log file (`meelog.txt`) corresponding to the time period of the session is saved as metadata. "0" = "don't save", which is assumed if the parameter is not specified.

`SESSION_INC_SMIB` – If this is "1", then the station static MIB is saved as metadata, as described in Section 3. "0" = "don't save", which is assumed if the parameter is not specified.

SESSION_INC_DES – If this is "1", then available and relevant design and calibration information is saved as metadata, as described in Section 3. "0" = "don't save", which is assumed if the parameter is not specified.

4.3 Observation Definitions

Each observation within a session is defined by the following keywords. Multiple observations are defined sequentially in the SDF.

4.3.1 Required Observation Keywords

OBS_ID – Observation Identification. This is intended to enable concise, unambiguous (in combination with PROJECT_ID and SESSION_ID) identification of the observation. Observation identification codes are sequentially-assigned integers, beginning with 1.

OBS_TITLE – This is redundant information given OBS_ID, but is included primarily for user convenience. Optional.

OBS_TARGET – This is intended to provide a convenient, standard place to indicate the intended "target" of the observation. This might be a specific source (e.g., "Cas A"), or might be used to indicate that this is an "All-Sky" (TBT/TBS) observation. This field is provided for the convenience of the observer only, and no specific format is required. In particular, it should be noted that this field is NOT used in any way by MCS to determine observing mode or pointing direction. Optional.

OBS_REMPI – Remarks from the PI on the observation. This is intended to convey information that might not be present or obvious through other session definition file parameters. Optional.

OBS_REMPO – Remarks from the Project Office on the observation. This is intended to convey information that might not be present or obvious through other session definition file parameters. Optional.

OBS_START_MJD – Modified Julian day (MJD) on which the observation is to start. Coordinated Universal Time (UTC) is assumed.

OBS_START_MPM – Time of day at which the observation is to start, written as integer milliseconds past UTC midnight (MPM). The range of values is 0 through 86399999; or, for days containing a leap second, 0 through 86400999.

OBS_START – Start time of the observation written in a format of the PI's choice. This is redundant information given OBS_START_MJD and OBS_START_MPM, but is included for user convenience. The suggested format is "UTC yyyy mm dd hh:mm:ss.sss". This parameter is not used by MCS.

OBS_DUR – Duration of the observation in milliseconds. This field is ignored for STEPPED mode (computed by tpss from step times), TBT mode (computed automatically from OBS_TBT_SAMPLES), and DIAG1 mode.

OBS_DUR+ – The duration of the observation written in a format of the PI's choice. This is redundant information given OBS_DUR (or OBS_TBT_SAMPLES) and is ignored by MCS, but is included for user convenience. The suggested format is "hh:mm:ss.sss". Optional; this parameter is not used by MCS.

OBS_MODE – Observing mode. One of: TRK_RADEC, TRK_SOL, TRK_JOV, TRK_LUN, STEPPED, TBT, TBS, or DIAG1.

OBS_BDM – Beam-dipole mode. When specified, MCS constructs a custom gain file that forms a beam using all stands in one polarization output while routing a single specified stand (the "dipole") to the other polarization output. The format of this string is "std gb gd pol", where std is the stand number for the dipole ($1 \leq$ std ≤ 256), gb is the gain applied to all other stands (the beam component), gd is the gain applied to the dipole stand, and pol is the polarization (X or Y). If not specified, the observation runs in normal beam-beam mode. A free-form string of up to 31 characters. Optional.

OBS_RA – Right ascension in decimal hours, epoch J2000. Required only for TRK_RADEC mode.

OBS_DEC – Declination in decimal degrees, epoch J2000. Required only for TRK_RADEC mode.

4.3.2 Beam, Tuning, and Bandwidth Keywords

OBS_B – Beam type. Meaningful only if OBS_MODE = TRK_RADEC, TRK_SOL, TRK_JOV, TRK_LUN, or STEPPED; should not appear otherwise. Options are:

- SIMPLE. Beamforming is by equalizing geometrical delays implied by the pointing direction and array geometry. MCS attempts to account for instrumental delays, gains, and phases through the system.
- HIGH_DR. Beamforming uses the same delay computation as SIMPLE, but the output data uses a higher bit depth to provide greater dynamic range.

If not specified, OBS_B will be set to SIMPLE. Note: SPEC_DELAYS_GAINS is valid only in step definitions (see OBS_STP_B below), not as an observation-level beam type.

OBS_FREQ1 – Center frequency of the first DRX tuning for beamforming modes or the center frequency for transient buffer – streaming (TBS) mode, expressed as an integer "tuning word". The center frequency in MHz is OBS_FREQ1 $\times$ $196/2^{32}$. For example, OBS_FREQ1 = 1073741824 corresponds to 49.000000 MHz. For beamforming modes, the valid range is 222417950–1928352663 (approximately 10.15–88.00 MHz). For TBS mode, the valid range is 65739295–2037918156 (approximately 3.00–93.00 MHz). This parameter is ignored if OBS_MODE = STEPPED (tuning frequencies are set per-step) or TBT.

OBS_FREQ1+ – Center frequency for the first DRX tuning or TBS center frequency, expressed in a format of the PI's choice. This is redundant information given OBS_FREQ1 and is ignored by MCS, but is included primarily for user convenience. The suggested format is "xx.xxxxxx MHz". Optional.

OBS_FREQ2 – Center frequency for the second DRX tuning, expressed as an integer "tuning word". See OBS_FREQ1 for additional details. A value of 0 disables the second tuning, resulting in a half-beam observation.

OBS_FREQ2+ – Center frequency for the second DRX tuning, expressed in a format of the PI's choice. See OBS_FREQ1+ for additional details. Optional.

OBS_BW – Bandwidth, expressed as an integer. For beamforming modes, an integer between 1 and 7. For TBS mode, an integer in the range 7–9. Refer to [1] for the mapping between bandwidth codes and sample rates.

`OBS_BW+` – Bandwidth, expressed in a format of the PI's choice. This is redundant information given `OBS_BW` and is ignored by MCS, but is included primarily for user convenience.

4.3.3 STEPPED Mode Keywords

The following keywords are required only for `STEPPED` mode observations:

`OBS_STP_N` – Number of steps in the observation.

`OBS_STP_RADEC` – Coordinate system for step pointing. 1 = RA/DEC (equatorial, J2000), 0 = azimuth/altitude (horizontal).

`OBS_STP_C1`[n] – For beam pointing direction at step n, this is either RA or AZ depending on `OBS_STP_RADEC`. Decimal hours or decimal degrees. $1 \leq n \leq$ `OBS_STP_N`.

`OBS_STP_C2`[n] – For beam pointing direction at step n, this is either DEC or ALT depending on `OBS_STP_RADEC`. Decimal degrees. $1 \leq n \leq$ `OBS_STP_N`.

`OBS_STP_T`[n] – Start time for this step in integer milliseconds from `OBS_START_MPM`. $1 \leq n \leq$ `OBS_STP_N`.

`OBS_STP_FREQ1`[n] – Center frequency for the first DRX tuning during step n, expressed as an integer "tuning word". The format is the same as used for `OBS_FREQ1`. $1 \leq n \leq$ `OBS_STP_N`.

`OBS_STP_FREQ1+`[n] – Center frequency for the first DRX tuning during step n, expressed in a format of the PI's choice. See `OBS_FREQ1+` for additional details. $1 \leq n \leq$ `OBS_STP_N`.

`OBS_STP_FREQ2`[n] – Center frequency for the second DRX tuning during step n, expressed as an integer "tuning word". The format is the same as used for `OBS_FREQ1`. $1 \leq n \leq$ `OBS_STP_N`.

`OBS_STP_FREQ2+`[n] – Center frequency for the second DRX tuning during step n, expressed in a format of the PI's choice. See `OBS_FREQ1+` for additional details. $1 \leq n \leq$ `OBS_STP_N`.

`OBS_STP_B`[n] – Beam type. $1 \leq n \leq$ `OBS_STP_N`. Options are:

- `SIMPLE`. (See `OBS_B` for additional details.)
- `HIGH_DR`. (See `OBS_B` for additional details.)
- `SPEC_DELAYS_GAINS`. Beamforming is by applying user-specified delays and gains. The delays and gains are specified by the parameters `OBS_BEAM_DELAY[][]` and `BEAM_-GAIN[][][][]`. MCS makes no attempt to account for instrumental delays and gains through the system.

`OBS_BEAM_DELAY`[n][p] – Corresponds to `BEAM_DELAY`[p] in [1]; this is the value during step n. Meaningful only when `OBS_STP_B`[n] = `SPEC_DELAYS_GAINS`, and must be specified in this case; otherwise optional. $1 \leq n \leq$ `OBS_STP_N`, $1 \leq p \leq 512$ (one per antenna, i.e., two per stand). Keywords must appear in sequential order by p.

`BEAM_GAIN`[n][p][q][r] – Corresponds to `BEAM_GAIN`[p][q][r] in [1]; this is the value during step n. Meaningful only when `OBS_STP_B`[n] = `SPEC_DELAYS_GAINS`, and must be specified in this case; otherwise optional. $1 \leq n \leq$ `OBS_STP_N`, $1 \leq p \leq 256$, $1 \leq q, r \leq 2$. Keywords must appear in sequential order by p, then q, then r.

Implementation Note

In a `STEPPED` observation, DRX settings (tuning frequencies, bandwidth, gain, and beam type) carry over from one step to the next. MCS only issues new commands to NDP when a setting changes relative to the previous step. This means that if consecutive steps share the same tuning or gain parameters, those parameters need not be re-specified in the SDF — the values from the preceding step remain in effect.

4.3.4 Optional Observation Parameters

The following observation parameters are optional, but allow additional control over the observation which may be useful in certain cases. If not specified, MCS will attempt to assign reasonable values.

`OBS_FEE[`n`][`p`]` – Controls power for the FEE on stand n, polarization p. "1" = "on", "0" = "off", "−1" = "MCS decides" (value used if not specified). $1 \leq n \leq 256$ and $p = 1$ or 2. n can also be 0, which is interpreted as meaning that this setting should apply for *all* n. Must be listed in order of increasing n. *NOTE: Because each FEE has only a single power input, both* `OBS_FEE` *settings for a stand must be set to "off" in order to have any effect.*

`OBS_ASP_FLT[`n`]` – Selects the "filter setting" for the ARX corresponding to stand n. This corresponds to the ASP MIB parameter "FIL". An integer in the range 0–7, or "−1" = "MCS decides" (value used if not specified). $1 \leq n \leq 256$. n can also be 0, which is interpreted as meaning that this setting should apply for *all* n. Must be listed in order of increasing n.

`OBS_ASP_AT1[`n`]` – Selects the first attenuator setting for the ARX corresponding to stand n. This corresponds to the ASP MIB parameter "AT1". This is an integer value between 0 and 15, or "−1" = "MCS decides" (value used if not specified). $1 \leq n \leq 256$. n can also be 0, which is interpreted as meaning that this setting should apply for *all* n. Must be listed in order of increasing n.

`OBS_ASP_AT2[`n`]` – Selects the second attenuator setting for the ARX corresponding to stand n. This corresponds to the ASP MIB parameter "AT2". This is an integer value between 0 and 15, or "−1" = "MCS decides" (value used if not specified). $1 \leq n \leq 256$. n can also be 0, which is interpreted as meaning that this setting should apply for *all* n. Must be listed in order of increasing n.

`OBS_ASP_AT3[`n`]` – Selects the third attenuator setting for the ARX corresponding to stand n. This corresponds to the ASP MIB parameter "AT3". This is an integer value between 0 and 31, or "−1" = "MCS decides" (value used if not specified). $1 \leq n \leq 256$. n can also be 0, which is interpreted as meaning that this setting should apply for *all* n. Must be listed in order of increasing n.

`OBS_TBT_SAMPLES` – Number of samples to acquire in a TBT observation. Default is 19600000. Maximum is 392000000. If not specified, the default value is used. This is only meaningful for `OBS_MODE` = `TBT`, and is ignored otherwise.

`OBS_DRX_GAIN` – This corresponds to the NDP DRX command parameter "DRX_GAIN". If both tunings should use the same gain, this is an integer value between 0 and 15. To specify different gains for the two tunings, this value is packed as $(\text{gain}_1 \times 16) + \text{gain}_2$, where gain_1 and gain_2 are each in the range 0–15, giving a combined range of 16–255. "−1" = "MCS decides" (value used if not specified).

Implementation Note

MCS currently does not act on the `OBS_FEE` and `OBS_ASP_*` parameters directly. These settings are recorded in the observation specification file and may be handled upstream by HAL (Heuristic Automation for LWA; see [5]).

5 Format of a Session Specification File

The session specification file (`.ses`) is a packed binary file. The following format specifiers are used: `s`n denotes a string of n bytes; `i1` and `i2` denote signed integers of 1 and 2 bytes, respectively, in little-endian byte order; `u2`, `u4`, and `u8` denote unsigned integers of 2, 4, and 8 bytes, respectively, in little-endian byte order.

Table 2 shows the format of the session specification file.

Table 2: Format of a session specification file.

Format	Field	Description
u2	FORMAT_VERSION	Format version of this file.
s9	PROJECT_ID	Project identifier (null-terminated string).
u4	SESSION_ID	Session identifier.
u2	SESSION_CRA	Configuration Request Authority.
i2	SESSION_DRX_BEAM	DRX beam assignment.
s32	SESSION_SPC	DR spectrometer configuration (see [2]).
u8	SESSION_START_MJD	Session start time, MJD (integer part).
u8	SESSION_START_MPM	Session start time, milliseconds past midnight.
u8	SESSION_DUR	Session duration in milliseconds.
u4	SESSION_NOBS	Number of observations in this session.
i2	SESSION_MRP_ASP	MIB recording period for ASP (minutes).
i2	SESSION_MRP_NDP	MIB recording period for NDP (minutes).
i2	SESSION_MRP_DR1	MIB recording period for DR1 (minutes).
i2	SESSION_MRP_DR2	MIB recording period for DR2 (minutes).
i2	SESSION_MRP_DR3	MIB recording period for DR3 (minutes).
i2	SESSION_MRP_DR4	MIB recording period for DR4 (minutes).
i2	SESSION_MRP_DR5	MIB recording period for DR5 (minutes).
i2	SESSION_MRP_SHL	MIB recording period for SHL (minutes).
i2	SESSION_MRP_MCS	MIB recording period for MCS (minutes).
i2	SESSION_MUP_ASP	MIB update period for ASP (minutes).
i2	SESSION_MUP_NDP	MIB update period for NDP (minutes).
i2	SESSION_MUP_DR1	MIB update period for DR1 (minutes).
i2	SESSION_MUP_DR2	MIB update period for DR2 (minutes).
i2	SESSION_MUP_DR3	MIB update period for DR3 (minutes).
i2	SESSION_MUP_DR4	MIB update period for DR4 (minutes).
i2	SESSION_MUP_DR5	MIB update period for DR5 (minutes).
i2	SESSION_MUP_SHL	MIB update period for SHL (minutes).
i2	SESSION_MUP_MCS	MIB update period for MCS (minutes).

Format	Field	Description
i1	SESSION_LOG_SCH	Save Scheduler log (0=no, 1=yes).
i1	SESSION_LOG_EXE	Save Executive log (0=no, 1=yes).
i1	SESSION_INC_SMIB	Include SSMIF (0=no, 1=yes).
i1	SESSION_INC_DES	Include design info (0=no, 1=yes).

6 Format of an Observation Specification File

The observation specification file (`.obs`) is a packed binary file. The following format specifiers are used: `f4` denotes a 4-byte floating-point number in little-endian (IEEE 754) format; `s`*n* denotes a string of *n* bytes; `i2` denotes a signed 2-byte integer in little-endian byte order; `u2`, `u4`, and `u8` denote unsigned integers of 2, 4, and 8 bytes, respectively, in little-endian byte order.

Table 3 shows the format of the observation specification file.

Table 3: Format of an observation specification file.

Format	Field	Description
u2	FORMAT_VERSION	Format version of this file.
s9	PROJECT_ID	Project identifier (null-terminated string).
u4	SESSION_ID	Session identifier.
i2	SESSION_DRX_BEAM	DRX beam assignment.
s32	SESSION_SPC	DR spectrometer configuration (see [2]).
u4	OBS_ID	Observation identifier.
u8	OBS_START_MJD	Observation start time, MJD (integer part).
u8	OBS_START_MPM	Observation start time, milliseconds past midnight.
u8	OBS_DUR	Observation duration in milliseconds.
u2	OBS_MODE	Observing mode (1=TRK_RADEC, 2=TRK_SOL, 3=TRK_JOV, 4=STEPPED, 7=DIAG1, 9=TRK_LUN, 10=TBT, 11=TBS).
s32	OBS_BDM	Beam-dipole mode configuration.
f4	OBS_RA	Right ascension (decimal hours, J2000).
f4	OBS_DEC	Declination (decimal degrees, J2000).
u2	OBS_B	Beam type (1=SIMPLE, 2=HIGH_DR; 0 for non-tracking modes).
u4	OBS_FREQ1	Tuning word for first tuning.
u4	OBS_FREQ2	Tuning word for second tuning.
u2	OBS_BW	Bandwidth code.
u4	OBS_STP_N	Number of steps (0 if not STEPPED).
u2	OBS_STP_RADEC	Step coordinate system (1=RA/DEC, 0=AZ/ALT).
var	Step Def. Blocks	OBS_STP_N step definition blocks (see below).
i2	OBS_FEE[1][1]	FEE setting, stand 1, pol 1.
	⋮	
i2	OBS_FEE[256][2]	FEE setting, stand 256, pol 2.

Format	Field	Description
i2	OBS_ASP_FLT[1]	ASP filter setting, stand 1.
	⋮	
i2	OBS_ASP_FLT[256]	ASP filter setting, stand 256.
i2	OBS_ASP_AT1[1]	ASP first attenuator, stand 1.
	⋮	
i2	OBS_ASP_AT1[256]	ASP first attenuator, stand 256.
i2	OBS_ASP_AT2[1]	ASP second attenuator, stand 1.
	⋮	
i2	OBS_ASP_AT2[256]	ASP second attenuator, stand 256.
i2	OBS_ASP_AT3[1]	ASP third attenuator, stand 1.
	⋮	
i2	OBS_ASP_AT3[256]	ASP third attenuator, stand 256.
u4	OBS_TBT_SAMPLES	Number of samples (TBT mode).
i2	OBS_DRX_GAIN	Gain (DRX/tracking/TBS modes).
u4	4294967295	Alignment check ($= 2^{32} - 1$).

6.1 Step Definition Block

Each step definition block (for step n) has the following format:

Table 4: Format of a step definition block.

Format	Field	Description
f4	OBS_STP_C1[n]	First coordinate (RA or azimuth).
f4	OBS_STP_C2[n]	Second coordinate (DEC or altitude).
u4	OBS_STP_T[n]	Step start time (ms from OBS_START_MPM).
u4	OBS_STP_FREQ1[n]	Tuning word for first tuning at step n.
u4	OBS_STP_FREQ2[n]	Tuning word for second tuning at step n.
u2	OBS_STP_B[n]	Beam type at step n (1=SIMPLE, 2=HIGH_DR, 3=SPEC_DELAYS_GAINS).
var	Beam Def. Block	Present only if OBS_STP_B[n] = SPEC_DELAYS_GAINS.
u4	4294967294	Alignment check ($= 2^{32} - 2$).

6.2 Beam Definition Block

A beam definition block is present within a step definition block only when OBS_STP_B[n] = SPEC_DELAYS_GAINS. Its format is:

Table 5: Format of a beam definition block.

Format	Field	Description
u2	OBS_BEAM_DELAY[n][1]	Beam delay, antenna 1.
	⋮	

Format	Field	Description
u2	OBS_BEAM_DELAY[n][512]	Beam delay, antenna 512.
i2	BEAM_GAIN[n][1][1][1]	Beam gain, stand 1, pol 1×1.
	⋮	
i2	BEAM_GAIN[n][256][2][2]	Beam gain, stand 256, pol 2×2.

Note

Note 1: The beam delay array has $2 \times 256 = 512$ entries (one per antenna, i.e., two per stand), while the beam gain array has $256 \times 2 \times 2 = 1024$ entries (a 2×2 Jones matrix per stand).

Note

Note 2: Fields that are not applicable to a given observing mode are set to 0. For example, `OBS_RA` and `OBS_DEC` are 0 for modes other than `TRK_RADEC`, and `OBS_STP_N` is 0 for modes other than `STEPPED`.

Implementation Note

The per-stand arrays in the observation specification file and beam definition blocks are always dimensioned for 256 stands, regardless of the actual number of stands at the station. For stations with fewer stands, the unused entries are set to their default values (-1 for optional parameters, 0 otherwise). This ensures that the binary file format is the same for all stations.

7 Format of a Session Metadata File

The session metadata file (`*_metadata.txt`) is a human-readable text file. One line is appended at the conclusion of each observation. Each line has the following format:

```
OBS_ID [OP_TAG] [BARCODE] OBS_OUTCOME [MSG]
```

where the fields are separated by whitespace and bracket-delimited as shown.

`OBS_ID` – The observation identifier, as defined previously.

`OP_TAG` – The name of the recording, derived from the MJD and a reference number (e.g., `060123_000456789`).

`BARCODE` – The barcode of the DRSU (Data Recording Storage Unit) on the data recorder used for this observation.

`OBS_OUTCOME` – An integer result code. Defined values are:

- **0 (`ME_OC_OK`):** Concluded successfully with no issues to report.
- **1 (`ME_OC_RCVD_ESF`):** Received a commanded “end session – failed” from inprocessing.
- **2 (`ME_OC_MEEI_STP`):** Session terminated as a result of a `STP` command.

`MSG` – Free-form text with additional remarks about the outcome.

> **Note**
>
> If a session fails before any observations complete, or if the session is ended by a failure command, a summary line is written with the `OP_TAG` field (without brackets) and no `BARCODE` field.

A Session Definition File Example

The following is an example of a session definition file (SDF). This example defines a session consisting of two short `TRK_RADEC` observations.

```
PI_ID 1
PI_NAME Ellingson, Steven
PROJECT_ID TPSS0001
PROJECT_TITLE Project Title
PROJECT_REMPI Project REMPI
PROJECT_REMPO Project REMPO

SESSION_ID 1
SESSION_TITLE tp_session_sch SDF test #1
SESSION_REMPI Test SDF consisting of 2 short TRK_RADEC observations
SESSION_REMPO Session REMPO

OBS_ID 1
OBS_TITLE Observation 1 Title
OBS_TARGET Observation 1 Target
OBS_REMPI Observation 1 REMPI
OBS_REMPO Observation 1 REMPO
OBS_START_MJD 55616
OBS_START_MPM 0
OBS_START UTC 2011 Feb 24 00:00:00.000
OBS_DUR 10000
OBS_DUR+ 00:00:10.000
OBS_MODE TRK_RADEC
OBS_RA 5.6
OBS_DEC +22.0
OBS_B SIMPLE
OBS_FREQ1 438261968
OBS_FREQ1+ 19.999999955 MHz
OBS_FREQ2 1928352663
OBS_FREQ2+ 87.999999977 MHz
OBS_BW 7
OBS_BW+ 19.6 MSPS (but not exactly sure what bandwidth this will be)

OBS_ID 2
OBS_TITLE Observation 2 Title
OBS_START_MJD 55616
OBS_START_MPM 10000
OBS_START UTC 2011 Feb 24 00:00:10.000
OBS_DUR 10000
OBS_DUR+ 00:00:10.000
```

```
OBS_MODE TRK_RADEC
OBS_RA 5.6
OBS_DEC +22.0
OBS_B SIMPLE
OBS_FREQ1 832697741
OBS_FREQ1+ 37.999999997 MHz
OBS_FREQ2 1621569285
OBS_FREQ2+ 73.999999990 MHz
OBS_BW 7
OBS_BW+ 19.6 MSPS (but not exactly sure what bandwidth this will be)
```

References

[**1**] J. Dowell, "Next-Generation Digital Processor (NDP) Interface Control Document," Ver. A, Long Wavelength Array Project, February 2026.

[**2**] C. Wolfe & J. Dowell, "Data Recorder (DR) Interface Control Document," Ver. 2.4.1, Long Wavelength Array Project, February 2026.

[**3**] S. Ellingson & J. Dowell, "LWA Station-Level Metadata," Ver. 10, LWA Engineering Memo MCS0031, February 2026.

[**4**] S. Ellingson, "Sensitivity of Antenna Arrays for Long-Wavelength Radio Astronomy," *IEEE Trans. Ant. & Prop.*, in press. Also available as LWA Memo 166, Dec 30, 2009.

[**5**] J. Dowell, "Heuristic Automation for LWA (HAL) Interface Control Document," Ver. A, Long Wavelength Array Project, February 2026.

[**6**] G. B. Taylor, J. Dowell, et al., "The Swarm Development Concept for the LWA," LWA Memo 210, Long Wavelength Array Project, 2019.