LWA Station-Level Observing Procedure and Associated Metadata
Ver. 1

Steve Ellingson*

June 16, 2010

Contents

1 Introduction **2**

2 Definitions **3**

3 Process **4**

4 Format of a Session Definition File **6**

5 Format of a Session Specification File **11**

6 Format of a Observation Specification File **12**

7 Format of a Session Metadata File **14**

A Session Definition File Example **15**

B Session Metadata File Example **16**

C Document History **17**

*Bradley Dept. of Electrical & Computer Engineering, 302 Whittemore Hall, Virginia Polytechnic Institute & State University, Blacksburg VA 24061 USA. E-mail: ellingson@vt.edu

1 Introduction

This memo documents the procedure for using the initial operational capability (IOC) monitoring and control system (MCS) to conduct observations with LWA-1. 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 DP and MCS-DR subsystems ([1] and [2], respectively).

2 Definitions

We begin with some definitions. For the purposes of this documentation, we shall refer to *projects*, which are said to consist of one or more *sessions*, which in turn are said to consist of one or more *observations*. Formal definitions of these terms follow:

- A **project** is a set of observations which collectively support a defined scientific objective. For LWA-1 post-IOC, a project will usually be an observing proposal which has been approved, possibly with changes or with additional information provided by the project office (PO).
- A **session** is a subset of *observations* (defined next) conducted within the scope of a *project* that is conducted using exactly one of the 5 principal DP outputs (that is, one of the 4 beam outputs or the TBW/TBN output) within a contiguous time duration, typically a few hours in length. Informally, a session might be described as an "observing run".
- An **observation** is henceforth defined as the process of collecting a *contiguous* block of data from the session-specified DP output, during which the parameters defining the observation (as documented in this memo) do not change. From the perspective of an LWA-1 user, observations are "atomic" units for scheduling purposes.

An observation is said to be conducted in one of the following **observing modes**. These are:

- RA/DEC Tracking (`TRK_RADEC`). This is a beam output mode in which the beam tracks a specified right ascension (RA) and declination (DEC).
- Solar Tracking (`TRK_SOL`). This is a beam output mode in which the beam tracks the Sun.
- Jupiter Tracking (`TRK_JOV`). This is a beam output mode in which the beam tracks Jupiter.
- Stepped Tracking (`STEPPED`). This is a beam output mode in which the beam is pointed to sequence of specified directions according to a predetermined schedule. The direction can be specified as either RA/DEC or AZ/EL. The beam is utterly fixed until the next scheduled repointing (so if RA/DEC is specified, the beam remains fixed at the corresponding AZ/EL until the next repointing). This mode can also be used to fix the beam for the entire duration of an observation (i.e., an observation consisting of just one "step"), to step *center frequency* as opposed to pointing, or to step in both center frequency and pointing in any combination.
- Transient Buffer – Wideband (`TBW`).
- Transient Buffer – Narrowband (`TBN`).

For example: A *project* might consist of 10 *sessions*, with each session corresponding to a single contiguous 4-hour time allocation of one beam (both tunings) on each of 10 different days. Let's say each session consists of observing 4 sources, and that the process for observing any given source is to point the beam at a strong calibrator while tuned to a given pair of frequencies (corresponding to the two tunings of the assigned DP beam output) for 10 minutes, and then to point the beam at the source for 50 minutes, and then to repeat the for a second different frequency pair. In this case, each session consists of 8 observations. All the observations are (probably) using the `TRK_RADEC` mode, but 8 combinations of RA/DEC and frequency pair are required. To further illustrate, it should be noted that in this case it is given that only one DP beam output was allocated; if however *two* DP beam outputs were allocated, then the project could be completed in one-half the elapsed time, using 2 concurrent sessions of four observations each, per day; total 20 sessions.

See the appendices for other examples of *sessions* and *observations*.

3 Process

In this section we outline the process for defining, scheduling, and conducting observations, and the associated input and output metadata. Note that MCS schedules *sessions*, not specifically observations. However, individual observations can be scheduled as sessions having only one observation.

1. The PI creates a *session definition file*. A session definition file is a text file which defines the session and its constituent observations. The format of a session definition file is specified in Section 4.

2. An operator submits the session definition file to MCS using `tp_session_sch`, a program which runs on MCS/TaskProcessor. Using `tp_session_sch`, the following happens:

   (a) The session definition file is parsed and checked for errors.

   (b) The derived information is displayed to the operator for confirmation.

   (c) The operator edits the session definition file "`REMPO`" fields (see Section 4) as appropriate. These are remarks pertaining to review and scheduling that the PO or operator wish to convey as metadata.

   (d) Specific resources are assigned to the observation. For example, it is at this point that the operator might assign a particular DP output beam to the session. Concurrently, `tp_session_sch` is checking for resource conflicts (e.g., making sure the assigned beam is not already allocated at the requested time, making sure requested ASP settings do not conflict with those of previously-scheduled observations, and so on).

   (e) `tp_session_sch` outputs the following files:

      - The session definition file, now modified as a result of Step 2c. The name of this file has the format `${PROJECT_ID}_${SESSION_ID}.txt`, where `${PROJECT_ID}` and `${SESSION_ID}` are the values associated with the parameters `PROJECT_ID` and `SESSION_ID` (defined in Section 4), respectively.
      - A *session specification file*. The name of this file has the format `${PROJECT_ID}_${SESSION_ID}.dat`. This is a packed binary file that specifies parameters that apply session-wide. These parameters are defined in Section 5.
      - One *observation specification file* for each defined observation. The filenames have the format `${PROJECT_ID}_${SESSION_ID}_${OBS_ID}.dat`, where `${OBS_ID}` is the value associated with the parameter `OBS_ID`, defined in Section 4. These are packed binary files that completely specify the scheduled observation. 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, and the primary DP data output is captured by MCS-DR.

4. As each observation concludes, MCS makes a modified copy of the observation specification file with the filename `${PROJECT_ID}_${SESSION_ID}_${OBS_ID}_${OBS_OUTCOME}.dat`, where `${OBS_OUTCOME}` is the value associated with the parameter `OBS_OUTCOME`, defined in Section 7. `${OBS_OUTCOME}` is an integer which indicates the outcome of the observation; such as whether the observation succeeded or failed. For parameters in the session definition file (and corresponding parameters in the input observation specification file) which are set to "MCS Decides", the contents of this file reflect the actual values used.

5. As the conclusion of the session, MCS creates a gzipped tarball with filename having the format `${PROJECT_ID}_${SESSION_ID}.tgz`. The tarball includes the following:

   - The session definition file, from Step 2e.
   - A modified version of the session specification file from Step 2e. The modifications are that all parameters are explicitly indicated, and for those which are (or would be) set to "MCS Decides", the actual values used are indicated.

- A *session metadata file*, with filename having the format `${PROGRAM_ID}_${SESSION_ID}_metadata.txt`. This is a human-readable text file reporting summary information about the session; in particular, information about the success or failure of each observation. The format of this file is given in Section 7.
- If `${SESSION_INC_SMIB}`= 1, the *station static MIB initialization file* is included. This file describes the "static" (unchanging) characteristics of the station, such as the array geometry, the installed components, and the connections between components. This is a human-readable text file whose format is documented in [3].
- A directory named `dynamic`, which contains various MIBs which are recorded during the session. The number and types of files in this directory depends on session definition file parameters having keywords beginning "`SESSION_MRP_`...". Files which may appear include the station dynamic MIB [3] and various subsystem MIBs (named according to their identifying three-letter acronyms). The root filenames of all files in this directory are suffixed with "*_MJD_MPM*" to indicate the mean julian date (MJD) and milliseconds past UT midnight (MPM) at which they were recorded.
- If `${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; it is edited to cover the time period of the session.
- Depending on the session definition file parameter `SESSION_INC_DES`, there may be a directory named `design` which contains various design data and possibly other information relevant to system calibration. See [3] for additional information.

The tarball described in item 5 above, plus the output observation definition files described in item 4 above, constitute the output metadata for the session.

4 Format of a Session Definition File

Session definition files 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 lines (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.

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 recommended that they appear in the session definition file. In each case, we give the identifying keyword, followed by information on valid values. We begin with parameters identifying 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 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 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 intended to be 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 intended to be convey information that might not be present or obvious through other session definition file parameters.

Next, keywords identifying the session:

- `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 intended to be 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 intended to be convey information that might not be present or obvious through other session definition file parameters.

Now, keywords defining an observation:

- 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.
- 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" (TBW/TBN) 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.
- OBS_REMPI: Remarks from the PI on the observation. This intended to be convey information that might not be present or obvious through other session definition file parameters.
- OBS_REMPO: Remarks from the Project Office on the observation. This intended to be convey information that might not be present or obvious through other session definition file parameters.
- OBS_START_MJD: Mean julian day (MJD) on which the observation is to start.
- OBS_START_MPM: Time of day at which the observation is to start, written as integer milliseconds past UT midnight (MPM).
- 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 "UT yyyy mm dd hh:mm:ss.sss". This parameter is not used by MCS.
- OBS_DUR: The duration of the observation in integer milliseconds. This parameter is ignored if OBS_MODE = TBW; see instead OBS_TBW_SAMPLES.
- 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_TBW_SAMPLES) and is ignored by MCS, but is included for user convenience. The suggested format is "hh:mm:ss.sss". This parameter is not used by MCS.
- OBS_MODE: Observing mode, as explained in Section 2. Valid entries here are: TRK_RADEC, TRK_SOL, TRK_JOV, STEPPED, TBW, and TBN.
- OBS_RA: RA (decimal) for beam pointing. Meaningful only if OBS_MODE = TRK_RADEC; otherwise this parameter is ignored.
- OBS_DEC: DEC (decimal) for beam pointing. Meaningful only if OBS_MODE = TRK_RADEC; otherwise this parameter is ignored.

OBS_BW	OBS_MODE = TRK_RADEC, TRK_SOL, TRK_JOV, or STEPPED	OBS_MODE = TBN	OBS_MODE = TBW
1	0.250 MSPS	1.000 kSPS	196 MSPS
2	0.500 MSPS	3.125 kSPS	196 MSPS
3	1.000 MSPS	6.250 kSPS	196 MSPS
4	2.000 MSPS	12.500 kSPS	196 MSPS
5	4.000 MSPS	25.000 kSPS	196 MSPS
6	9.800 MSPS	50.000 kSPS	196 MSPS
7	19.600 MSPS	100.000 kSPS	196 MSPS

Table 1: Relationship between the parameter OBS_BW and sample rate, adapted from information in the DP ICD [1]. Note that the DP ICD does not specify filter shapes; however bandwidth (e.g., 3 dB bandwidth) for modes other than TBW can be inferred to be slightly less than the indicated sample rate; for TBW it is determined by ARX configuration.

- OBS_B: Beam type. Meaningful only if OBS_MODE = TRK_RADEC, TRK_SOL, or TRK_JOVE. 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.
  - MAX_SNR. Delays are selected so as to maximize beam sensitivity. This is known to result in significant improvement in sensitivity over SIMPLE beamforming [4], but may have undesirable effects on main lobe shape or sidelobe characteristics.
- OBS_FREQ1: Center frequency for the first DRX tuning, expressed as an integer "tuning word". The center frequency will be OBS_FREQ1 $\times 196/2^{32}$ MHz. For example, OBS_FREQ1 = 1073741824 corresponds to a center frequency of 49.000000 MHz. If OBS_MODE = STEPPED, this parameter is ignored.
- OBS_FREQ1+: Center frequency for the first DRX tuning, 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".
- OBS_FREQ2: Center frequency for the first DRX tuning, expressed as an integer "tuning word". See OBS_FREQ1 for additional details.
- OBS_FREQ2+: Center frequency for the first DRX tuning, expressed in a format of the PI's choice. See OBS_FREQ1+ for additional details.
- OBS_BW: Bandwidth, expressed as an integer between 1 and 7. Refer to Table 1 for more information.
- 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.

Continuing with observation definition, the following keywords are meaningful only for OBS_MODE = STEPPED, and are ignored otherwise:

- OBS_STP_N: Number of steps, written as an integer.
- OBS_STP_RADEC: Coordinate system for the parameters OBS_STP_C1 and OBS_STP_C2. "1" means RA/DEC, "0" means AZ/EL.
- OBS_STP_C1[n]: For beam pointing direction at step n, this is either RA or AZ depending on OBS_STP_RADEC. Decimal. $1 \leq n \leq$ OBS_STP_N.

- OBS_STP_C2[n]: For beam pointing direction at step n, this is either DEC or EL depending on OBS_STP_RADEC. Decimal. $1 \leq n \leq$ OBS_STP_N.
- OBS_STP_T[n]: Start time for this step in integer milliseconds from OBS_START_MPM. Valid values are between 0 and OBS_DUR-5. $1 \leq n \leq$ 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 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 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_STP_B for additional details.)
  - MAX_SNR. (See OBS_STP_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 OBS_BEAM_GAIN[][][][]. MCS makes no attempt to account for instrumental delays and gains through system.

The following 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. First parameters that have session-wide scope:

- SESSION_DRX_BEAM: This is an integer 1-4 which can be used to request the corresponding DP beam output channel. The value "-1" = "MCS decides" (value used if not specified). This is not meaningful if OBS_MODE = TBN or TBW, and is ignored for those observing modes.
- SESSION_MRP_*sss*: This sets the recording period for the MIB associated with the level-1 subsystem *sss*, where *sss* is the usual three-letter acronym (e.g., "ASP", "DP_", etc.). Integer minutes. For example: SESSION_MRP_ASP = 5 will cause MCS to archive (record) a copy of the ASP MIB every 5 minutes for the duration of the observation. The recorded MIB files are then available as metadata following the observation. "0" = "never record", and "-1" = "MCS decides". 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*.
- SESSION_MUP_*sss*: This requests a minimum update period for the MIB associated with the level-1 subsystem *sss*, where *sss* is the usual three-letter acronym (e.g., "ASP", "DP_", etc.). 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". 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.
- 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. This is assumed if not specified; otherwise, use "0" = "don't save".
- 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. This is assumed if not specified; otherwise, use "0" = "don't save".

- `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.

The following optional parameters apply to observations:

- `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). $1 \leq n \leq 260$ and $p = 1$ or 2. n can also be 0, which is interpreted as meaning that this setting should apply for *all* n.
- `OBS_ASP_FLT[`n`]`: Selects the “filter setting” for the ARX corresponding to stand n. This corresponds to the ASP MIB parameter “FIL”. “0” = “split”, “1” = “full”, “2” = “reduced”, “3” = “off”, and “-1” = “MCS decides” (value used if not specified). $1 \leq n \leq 260$. n can also be 0, which is interpreted as meaning that this setting should apply for *all* 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). $1 \leq n \leq 260$. n can also be 0, which is interpreted as meaning that this setting should apply for *all* 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). $1 \leq n \leq 260$. n can also be 0, which is interpreted as meaning that this setting should apply for *all* n.
- `OBS_ASP_ATS[`n`]`: Selects the split attenuator setting for the ARX corresponding to stand n. This corresponds to the ASP MIB parameter “ATS”. This is an integer value between 0 and 15, or “-1” = “MCS decides” (value used if not specified). $1 \leq n \leq 260$. n can also be 0, which is interpreted as meaning that this setting should apply for *all* n.
- `OBS_TBW_BITS`: Number of bits per TBW output sample. Valid values are 12 (value used if not specified) and 4. This is only meaningful for `OBS_MODE` = `TBW`, and is ignored otherwise.
- `OBS_TBW_SAMPLES`: Number of samples to acquire in a TBW observation. The largest valid values are 12000000 for `OBS_TBW_BITS`= 12 and 36000000 for `OBS_TBW_BITS`= 4. If not specified, the largest valid values are used. This is only meaningful for `OBS_MODE` = `TBW`, and is ignored otherwise.
- `OBS_TBN_GAIN`: This corresponds to the DP TBN command parameter “TBW_GAIN”. This is an integer value between 0 and 15, or “-1” = “MCS decides” (value used if not specified).
- `OBS_DRX_GAIN`: This corresponds to the DP DRX command parameter “DRX_GAIN”. This is an integer value between 0 and 15, or “-1” = “MCS decides” (value used if not specified).
- `OBS_BEAM_DELAY[`n`][`p`]`: Corresponds to `BEAM_DELAY[`p`]` in [1]; this is the value during step n. Meaningful only when `OBS_MODE` = `STEPPED` and `OBS_STP_B[`n`]` = `SPEC_DELAYS_GAINS`, and must be specified in this case. $1 \leq n \leq$ `OBS_STP_N`, $1 \leq p \leq 520$.
- `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_MODE` = `STEPPED` and `OBS_STP_B[`n`]` = `SPEC_DELAYS_GAINS`, and must be specified in this case. $1 \leq n \leq$ `OBS_STP_N`, $1 \leq p \leq 260$, $1 \leq q, r \leq 2$.

5 Format of a Session Specification File

The session specification file is a packed binary file which indicates parameters used by MCS/Executive that apply session-wide. Note that if this is the post-observation file, any values indicating "MCS Decides" in the pre-observation file will be replaced with actual value used during the session (i.e., what MCS decided).

The format of the file is as shown below. The first column is a format specifier, which is decoded as shown below:

s*n*: character string, *n* bytes, right-padded with "\0" characters.

u2: unsigned integer, 2 bytes, little-endian byte order.

u4: unsigned integer, 4 bytes, little-endian byte order.

The second column indicates the parameter keyword. The third column defines any parameter values not previously defined, or provides additional comments as necessary.

u2	`FORMAT_VERSION`	Number identifying this version of the file format. (Same as version number of this document.)
s9	`PROJECT_ID`	
u4	`SESSION_ID`	
u2	`SESSION_DRX_BEAM`	
u2	`SESSION_MRP_ASP`	
u2	`SESSION_MRP_DP_`	
u2	`SESSION_MRP_DR1`	
u2	`SESSION_MRP_DR2`	
u2	`SESSION_MRP_DR3`	
u2	`SESSION_MRP_DR4`	
u2	`SESSION_MRP_DR5`	
u2	`SESSION_MRP_SHL`	
u2	`SESSION_MUP_ASP`	
u2	`SESSION_MUP_DP_`	
u2	`SESSION_MUP_DR1`	
u2	`SESSION_MUP_DR2`	
u2	`SESSION_MUP_DR3`	
u2	`SESSION_MUP_DR4`	
u2	`SESSION_MUP_DR5`	
u2	`SESSION_MUP_SHL`	
u2	`SESSION_LOG_SCH`	
u2	`SESSION_LOG_EXE`	
u2	`SESSION_INC_SMIB`	
u2	`SESSION_INC_DES`	

6 Format of a Observation Specification File

The observation specification file is a packed binary file which completely describes an observation, save any session-wide parameters defined in the session specification file. Note that if this is the post-observation file, any values indicating "MCS Decides" in the pre-observation file will be replaced with the actual value used during the observation (i.e., what MCS decided). The same applies if for any other reason it is necessary to use settings different from those indicated in the input observation specification file.

The format of the file is as shown below. The first column is a format specifier, which is decoded as shown below:

- f4: floating point, 4 bytes, little-endian byte order.
- s*n*: character string, *n* bytes, right-padded with "\0" characters.
- i2: signed integer, 2 bytes, little-endian byte order.
- u2: unsigned integer, 2 bytes, little-endian byte order.
- u4: unsigned integer, 4 bytes, little-endian byte order.

The second column indicates the parameter keyword. The third column defines any parameter values not previously defined, or provides additional comments as necessary.

u2	`FORMAT_VERSION`	Number identifying this version of the file format. (Same as version number of this document.)
s9	`PROJECT_ID`	
u4	`SESSION_ID`	
u4	`OBS_ID`	
u4	`OBS_START_MJD`	
u4	`OBS_START_MPM`	
u4	`OBS_DUR`	If `OBS_MODE` = 5 (`TBW`), this will be 0.
u2	`OBS_MODE`	1=`TRK_RADEC`, 2=`TRK_SOL`, 3=`TRK_JOV`, 4=`STEPPED`, 5=`TBW`, 6=`TBN`.
f4	`OBS_RA`	If `OBS_MODE` $\neq$ 1 (`TRK_RADEC`), this will be 0.
f4	`OBS_DEC`	If `OBS_MODE` $\neq$ 1 (`TRK_RADEC`), this will be 0.
u2	`OBS_B`	1=`SIMPLE`, 2=`MAX_SNR`; If `OBS_MODE` > 3, this will be 0.
u4	`OBS_FREQ1`	If `OBS_MODE` = 4 (`STEPPED`), this will be 0.
u4	`OBS_FREQ2`	If `OBS_MODE` = 4 (`STEPPED`), this will be 0.
u2	`OBS_BW`	If `OBS_MODE` = 5 (`TBW`), this will be 0.
u4	`OBS_STP_N`	If `OBS_MODE` $\neq$ 4 (`STEPPED`), this will be 0.
u2	`OBS_STP_RADEC`	If `OBS_MODE` $\neq$ 4 (`STEPPED`), this will be 0.
var	*Step Def. Blocks*	A total of `OBS_STP_N` step definition blocks (see below).
i2	`OBS_FEE[  1][1]`	See Note 1.
i2	`OBS_FEE[  1][2]`	
i2	`OBS_FEE[  2][1]`	
i2	`OBS_FEE[  2][2]`	
	...	
i2	`OBS_FEE[260][1]`	
i2	`OBS_FEE[260][2]`	
i2	`OBS_ASP_FLT[  1]`	
i2	`OBS_ASP_FLT[  2]`	
	...	
i2	`OBS_ASP_FLT[260]`	
i2	`OBS_ASP_AT1[  1]`	
i2	`OBS_ASP_AT1[  2]`	
	...	

i2	OBS_ASP_AT1[260]	
i2	OBS_ASP_AT2[1]	
i2	OBS_ASP_AT2[2]	
	...	
i2	OBS_ASP_AT2[260]	
i2	OBS_ASP_ATS[1]	
i2	OBS_ASP_ATS[2]	
	...	
i2	OBS_ASP_ATS[260]	
u2	OBS_TBW_BITS	If OBS_MODE $\neq$ TBW, this will be 0.
u4	OBS_TBW_SAMPLES	If OBS_MODE $\neq$ TBW, this will be 0.
i2	OBS_TBN_GAIN	If OBS_MODE $\neq$ TBW, this will be 0. Also, see Note 1.
i2	OBS_DRX_GAIN	If OBS_MODE $>$ 4 (STEPPED), this will be 0. Also, see Note 1.
u4	4294967295	$= 2^{32} - 1$, used as a known value to confirm byte alignment when reading file.

The n^{th} *step definition block* is defined as follows ($1 \leq n \leq$OBS_STP_N):

f4	OBS_STP_C1[n]	
f4	OBS_STP_C2[n]	
u4	OBS_STP_T[n]	
u4	OBS_STP_FREQ1[n]	
u4	OBS_STP_FREQ2[n]	
u2	OBS_STP_B[n]	1=SIMPLE, 2=MAX_SNR, 3=SPEC_DELAYS_GAINS
var	*Beam Def. Block*	Appears if OBS_STP_B[n] = 3 (SPEC_DELAYS_GAINS).
u4	4294967294	$= 2^{32} - 2$, used as a known value to confirm byte alignment when reading file.

A *beam definition block* is defined as follows. See [1] for additional details.

u2	OBS_BEAM_DELAY[n][1]	Corresponds to BEAM_DELAY[] in [1].
u2	OBS_BEAM_DELAY[n][2]	
	...	
u2	OBS_BEAM_DELAY[n][520]	
i2	BEAM_GAIN[n][1][1][1]	Corresponds to BEAM_GAIN[][][] in [1].
i2	BEAM_GAIN[n][1][1][2]	
i2	BEAM_GAIN[n][1][2][1]	
i2	BEAM_GAIN[n][1][2][2]	
i2	BEAM_GAIN[n][2][1][1]	
i2	BEAM_GAIN[n][2][1][2]	
i2	BEAM_GAIN[n][2][2][1]	
i2	BEAM_GAIN[n][2][2][2]	
	...	
i2	BEAM_GAIN[n][260][1][1]	
i2	BEAM_GAIN[n][260][1][2]	
i2	BEAM_GAIN[n][260][2][1]	
i2	BEAM_GAIN[n][260][2][2]	

Code at beginning of OBS_COMMENT value field	Meaning
0	Remarks follow
Others TBD	*TBD*

Table 2: Meaning of codes appearing in OBS_COMMENT lines.

7 Format of a Session Metadata File

The session metadata file is a human-readable text file, identical in structure to the session definition file, but containing only a subset of lines from the session definition file. These lines are as indicated by the following keyword list:

PI_ID

PI_NAME

PROJECT_ID

PROJECT_TITLE

SESSION_ID

SESSION_TITLE

Then for each observation:

OBS_ID

OBS_TITLE

OBS_TARGET

OBS_OUTCOME

All parameters are as defined previously; however OBS_OUTCOME is new. This is assigned by MCS to indicate the degree to which the observation was successful. Possible values are:

0, indicating that the observation executed with no noted problems sufficient to warrant concern.

1, indicating a *possible* problem with data, or a problem that has affected metadata integrity.

2, indicating a problem that is certain to have affected data.

3, indicating that the observation terminated or failed before the scheduled end time.

4, indicating that the observation did not execute.

In the case that OBS_OUTCOME $\neq 0$, one or more additional lines appear at the end of the set of lines for this observation. The keyword for each line is OBS_COMMENT. The value will be one of the codes from Table 2, possibly followed by one or more whitespace characters and additional explanatory text.

A Session Definition File Example

Below is shown the contents of an sample session definition file. If this file were the output of `tp_session_sch`, it would be given the name `LQ041_3.txt`.

```
PI_ID         32
PI_NAME       Jones, Pat

PROJECT_ID    LQ041
PROJECT_TITLE Observations of Astrophysical Sources of Radio Emission
PROJECT_REMPI In place of this text would be additional notes from the PI that would
be useful to have carried along as metadata.
PROJECT_REMPO Here are some comments from the project office; e.g., additional admin
info.

SESSION_ID    3
SESSION_TITLE Crab Pulsar at 20, 38, 74, and 88 MHz
SESSION_REMPI Using my two tunings first for 20 and 88 MHz, then 38 and 74 MHz.
SESSION_REMPO Here are some comments from the project office; e.g., additional tech
info.

OBS_ID        1
OBS_TITLE     20 MHz and 88 MHz
OBS_TARGET    B0531+21
OBS_REMPI     In place of this text would be additional notes from the PI that would
be useful to have carried along as metadata
OBS_REMPO     Here are some comments from the project office
OBS_START_MJD 54828
OBS_START_MPM 12345698
OBS_START     UT 2011 Apr 22 14:10:12.698
OBS_DUR       3600000
OBS_DUR+      01:00.000
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     38 MHz and 74 MHz
OBS_START_MJD 54828
OBS_START_MPM 127056789
OBS_START     UT 2011 Apr 22 15:10:12.698
OBS_FREQ1     832697741
OBS_FREQ1+    37.999999997 MHz
OBS_FREQ2     1621569285
OBS_FREQ2+    73.999999990 MHz
```

B Session Metadata File Example

Below is shown the contents of an sample observation metadata file, based on the session definition file example given in Appendix A. This file would be given the name `LQ041_3_metadata.txt`.

```
PI_ID          32
PI_NAME        Jones, Pat
PROJECT_ID     LQ041
PROJECT_TITLE  Observations of Astrophysical Sources of Radio Emission
SESSION_ID     3
SESSION_TITLE  Crab Pulsar at 20, 38, 74, and 88 MHz
OBS_ID         1
OBS_TITLE      20 MHz and 88 MHz
OBS_TARGET     B0531+21
OBS_OUTCOME    0
OBS_ID         1
OBS_TITLE      38 MHz and 74 MHz
OBS_TARGET     B0531+21
OBS_OUTCOME    0
```

C Document History

- Version 1 (June 16, 2010): First version.

References

[1] M. Soriano, "Interface Control Document for Digital Signal Processing Subsystem (DP)," Ver. H, Long Wavelength Array Internal Report, Dec 15, 2009.

[2] C. Wolfe, S. Ellingson, & C. Patterson, "Interface Control Document for Monitor and Control System Data Recorder (MCS-DR)," Ver. 1.0, Long Wavelength Array Engineering Memo MCS0025, Mar 22, 2010.

[3] S. Ellingson, "LWA Station-Level Metadata," Ver. 1, Long Wavelength Array Engineering Memo MCS0031, June 16, 2010.

[4] S. Ellingson, "Sensitivity of Antenna Arrays for Long-Wavelength Radio Astronomy", *IEEE Trans. Ant. & Prop.*, *in press*. Also available as Long Wavelength Array Memo 166, Dec 30, 2009, at http://www.phys.unm.edu/∼lwa/memos.