LWA Station-Level Metadata

Version 10

S. Ellingson, J. Dowell

February 2026

Long Wavelength Array Project

Change Record

Version	Date	Description
1	June 16, 2010	First version.
3	Feb 27, 2011	GEO_EL field added; RPD_GAIN[m] deprecated, replaced by RPD_A0[m], RPD_A1[m], RPD_FREF[m], and RPD_STR[m]; For many indexed params, added ability to define default value without index; MCS_CRA field added.
4	March 29, 2011	Requirements imposed on ordering of keywords in SSMIF; "Format of the Station Dynamic MIB" section completely revised (now C struct instead of dbm); Various small revisions and clarifications.
5	April 13, 2011	For DP, PWR_NAME values now include DC1, DC2, FAN, SYN, and SWI; "settings" structure added to station dynamic MIB; Added example of SSMIF (text) file; Added keywords MRP_sss, MUP_sss, FEE[], ASP_FLT[], ASP_AT1[], ASP_AT2[], ASP_ATS[], TBN_GAIN[], DRX_GAIN[] to SSMIF.
10	Feb 2026	DP1/DP2 boards replaced by SNAP boards and Servers (NDP); DR_DP renamed to DR_NDP; DR_SHLF removed; ASP_ATS replaced by ASP_AT3; TBN_GAIN removed; Added array pointing correction parameters (PC_AXIS_TH, PC_AXIS_PH, PC_ROT); N_STD capped at 256; SDM updated to match (NDP, SNAP, Server status fields); All DP_ references changed to NDP; Removed incorrect sentinel values from FEE_GAI2 and FEE_ANT2 descriptions.

Contents

1 Introduction

This memo documents the format of station-level metadata for an LWA station's monitoring and control system (MCS). The metadata described in this document consists of "station static" and "station dynamic" information, which is distinct from the observation-related metadata described in [1].

2 Format of a Station Static MIB Initialization File

Initialization files are human-readable text files. Each line of a file has one of the following formats:

- `keyword data # comment`
- `# comment`
- empty line

where `keyword` identifies a parameter and contains no internal whitespace, `data` consists of printable non-whitespace characters (not including "#"), and `comment` is preceded by "#" and may contain any printable characters including spaces.

Lines may be up to 4096 characters in length. Empty lines are allowed and their use is encouraged for readability.

Parameters must appear in the order listed below. For each parameter, the keyword, definition, and valid values are given.

2.1 Station-Level Parameters

`FORMAT_VERSION`
: Integer equal to the version number of this document (i.e., the format version). Included to account for the possibility of format modifications over time. This document describes version 10 of the Station Static MIB Initialization File (SSMIF).

`STATION_ID`
: Two-letter station identification code, intended to enable concise, unambiguous identification of the station.

`GEO_N`
: WGS84 latitude of the origin of the station's local coordinate system. Decimal degrees, with North and South being indicated as "+" and "−", respectively. This position is an arbitrarily-selected reference point and may not necessarily correspond to the location of the phase center of the station during an observation.

`GEO_E`
: WGS84 longitude of the origin of the station's local coordinate system. Decimal degrees, with East and West being indicated as "+" and "−", respectively. This position is an arbitrarily-selected reference point and may not necessarily correspond to the location of the phase center of the station during an observation.

`GEO_EL`
: Elevation (above mean sea level) of the origin of the station's local coordinate system, in

meters. This position is an arbitrarily-selected reference point and may not necessarily correspond to the location of the phase center of the station during an observation.

`N_STD`
: Maximum number of stands; expected to be ≤ 256.

2.2 Stand Locations

Stand locations are specified by stand ID n, where $1 \leq n \leq$ `N_STD`. Stands must appear in order.

`STD_LX[`n`]`
: x coordinate in meters of the feedpoints of stand n ($1 \leq n \leq$ `N_STD`) in the local coordinate system. The $+x$ direction points East.

`STD_LY[`n`]`
: y coordinate in meters of the feedpoints of stand n ($1 \leq n \leq$ `N_STD`) in the local coordinate system. The $+y$ direction points North.

`STD_LZ[`n`]`
: z coordinate in meters of the feedpoints of stand n ($1 \leq n \leq$ `N_STD`) in the local coordinate system. The $+z$ direction points to the Zenith.

2.3 Antenna Parameters

Antenna parameters are indexed by antenna number n, where $1 \leq n \leq 2 \times$ `N_STD`.

`ANT_STD[`n`]`
: The stand on which antenna n ($1 \leq n \leq 2 \times$ `N_STD`) is mounted. This will be set to $\lfloor (n-1)/2 \rfloor + 1$ if not otherwise specified.

`ANT_ORIE[`n`]`
: The intended orientation (polarization) of antenna n ($1 \leq n \leq 2 \times$ `N_STD`). 0 = "intended to be North–South"; 1 = "intended to be East–West". This will be set to $(n-1) \bmod 2$ if not otherwise specified.

`ANT_STAT[`n`]`
: The status of antenna n ($1 \leq n \leq 2 \times$ `N_STD`). See Note 1. This will be set to 3 ("OK") if not otherwise specified.

`ANT_THETA[`n`]`
: The undesired rotation [deg] of the North or East arm of antenna n ($1 \leq n \leq 2 \times$ `N_STD`) in the elevation plane, relative to nominal (0°). Positive sign means increasing angle with respect to the $+z$-axis of the local coordinate system, in the direction of the $+x$ axis. Will be set to 0.0 (no error) if not otherwise specified.

`ANT_PHI[`n`]`
: The undesired rotation [deg] of the North or East arm of antenna n ($1 \leq n \leq 2 \times$ `N_STD`) in the azimuth plane, relative to nominal (0°). Positive sign means increasing angle with respect to the $+x$-axis of the local coordinate system in the direction of the $+y$ axis. Will be set to 0.0 (no error) if not otherwise specified.

`ANT_DESI[`n`]`
: An integer code which identifies the design of antenna n ($1 \leq n \leq 2 \times$ `N_STD`). See Note 2.

Design information expected to be indexed by this code includes the mechanical specification (specific design/manufacture/model), complex vector effective length vs. frequency and pattern direction, and self-impedance vs. frequency. This will be set to "1" unless otherwise specified. Use "0" to indicate a different but unknown/undocumented design. `ANT_DESI` (without `[`n`]`) will result in `ANT_DESI[`n`]` being set to `ANT_DESI` for all n; although subsequent uses of `ANT_DESI[`n`]` can override this for selected n.

2.4 FEE Parameters

`N_FEE`
: Number of FEEs to be described in this file; expected to be ≤ 256.

`FEE_ID[`m`]`
: Label or serial number which unambiguously identifies FEE m ($1 \leq m \leq$ `N_FEE`). Limited to 10 characters. Default: `"UNK"`.

`FEE_STAT[`m`]`
: The status of FEE m ($1 \leq m \leq$ `N_FEE`). See Note 1. This will be set to 3 ("OK") unless otherwise specified.

`FEE_DESI[`m`]`
: An integer code which identifies the design of FEE m ($1 \leq m \leq$ `N_FEE`). See Note 2. Design information expected to be indexed by this code includes electrical and mechanical descriptions and frequency-domain transfer function described as (a) coefficients in a polynomial fit (representative of all FEEs with this design code) and (b) measurements of a representative FEE. This will be set to "1" unless otherwise specified. Use "0" to indicate a different but unknown/undocumented design. `FEE_DESI` (without `[`m`]`) will result in `FEE_DESI[`n`]` being set to `FEE_DESI` for all n; although subsequent uses of `FEE_DESI[`n`]` can override this for selected n.

`FEE_GAI1[`m`]`
: Gain [dB] of FEE m ($1 \leq m \leq$ `N_FEE`) port 1, assuming nominal input and output terminations, at the reference frequency of 38 MHz. This will be set to 35.7 unless otherwise specified. `FEE_GAI1` (without `[`m`]`) will result in `FEE_GAI1[`n`]` being set to `FEE_GAI1` for all n; although subsequent uses of `FEE_GAI1[`n`]` can override this for selected n.

`FEE_GAI2[`m`]`
: Gain [dB] of FEE m ($1 \leq m \leq$ `N_FEE`) port 2, assuming nominal input and output terminations, at the reference frequency of 38 MHz. If this FEE has only one port, then this should be -200. This will be set to 35.7 unless otherwise specified. `FEE_GAI2` (without `[`m`]`) will result in `FEE_GAI2[`n`]` being set to `FEE_GAI2` for all n; although subsequent uses of `FEE_GAI2[`n`]` can override this for selected n.

`FEE_ANT1[`m`]`
: Antenna to which port 1 of FEE m ($1 \leq m \leq$ `N_FEE`) is connected. Normally in the range 1 to $2 \times$ `N_STD`. A value of 0 means the FEE input is open-circuited. If not specified, then `FEE_ANT1[1]` will be 1, `FEE_ANT2[1]` will be 2, `FEE_ANT1[2]` will be 3, `FEE_ANT2[2]` will be 4, and so on.

`FEE_ANT2[`m`]`
: Antenna to which port 2 of FEE m ($1 \leq m \leq$ `N_FEE`) is connected. Normally in the range 1

to $2 \times$ N_STD. A value of 0 means the FEE input is open-circuited or has only one port. See FEE_ANT1[m] (above) for default ordering.

2.4.1 FEE Power Sources

FEE power sources must be identified in order of FEE index m, with each FEE's source being specified using FEE_RACK[m] and FEE_PORT[m] keywords, in that order.

FEE_RACK[m]
: From the perspective of SHL, this is the rack (1 to N_PWR_RACK) in which the power supply powering this FEE is located. A value of 0 means this parameter is unknown. This parameter is used in conjunction with FEE_PORT[m] to identify the power source for this FEE.

FEE_PORT[m]
: From the perspective of SHL, this is the power port corresponding to the power supply powering this FEE. A value of 0 means this parameter is unknown. This parameter is used in conjunction with FEE_RACK[m] to identify the power source for this FEE.

2.5 Cable (RPD) Parameters

N_RPD
: Maximum number of cables connecting FEEs to SEP; expected to be ≤ 512.

RPD_ID[m]
: Label or tag which unambiguously identifies cable m ($1 \leq m \leq$ N_RPD). Maximum 25 characters. Default: "UNK".

RPD_STAT[m]
: The status of cable m ($1 \leq m \leq$ N_RPD). See Note 1. Set to 3 ("OK") unless otherwise specified.

RPD_DESI[m]
: An integer code which identifies the design of cable m ($1 \leq m \leq$ N_RPD). See Note 2. Design information expected to be indexed by this code includes cable type, electrical and mechanical descriptions, frequency-domain transfer function described as coefficients in a polynomial fit (representative of all cables with this design code). Set to "1" unless otherwise specified. Use "0" to indicate that design is unknown or undocumented. *The value "2" has been used for the (primarily) LMR-400 cables to Stand 258.* RPD_DESI (without [m]) will result in RPD_DESI[n] being set to RPD_DESI for all n; although subsequent uses of RPD_DESI[n] can override this for selected n.

RPD_LENG[m]
: Length [m] of cable m ($1 \leq m \leq$ N_RPD). Set to 0.0 unless otherwise specified.

The following cable parameters appear in the order shown. First the default-setting keyword (without index) is given, then the per-cable values: RPD_VF, RPD_DD, RPD_A0, RPD_A1, RPD_FREF, RPD_STR, followed by RPD_VF[m], RPD_DD[m], RPD_A0[m], RPD_A1[m], RPD_FREF[m], RPD_STR[m].

RPD_VF[m]
: Velocity factor [%] of cable m ($1 \leq m \leq$ N_RPD) at the reference frequency of 10 MHz. Set

to 83 unless specified otherwise. RPD_VF (without [m]) will result in RPD_VF[n] being set to RPD_VF for all n; although subsequent uses of RPD_VF[n] can override this for selected n.

RPD_DD[m]
: Dispersive delay [ns] of cable m ($1 \leq m \leq$ N_RPD) at the reference frequency of 10 MHz and reference length of 100 m. This is the additional propagation time beyond that expected by dividing length by (velocity factor $\times$ the speed of light in free space) due to cable dispersion. Set to 2.4 unless specified otherwise. RPD_DD (without [m]) will result in RPD_DD[n] being set to RPD_DD for all n; although subsequent uses of RPD_DD[n] can override this for selected n.

RPD_A0[m]
: α_0 [m^{-1}] of cable m ($1 \leq m \leq$ N_RPD) at the reference frequency RPD_FREF[m]. This is used to calculate cable gain given length and frequency via the Memo 170 model. Set to 0.00428 unless otherwise specified. RPD_A0 (without [m]) will result in RPD_A0[n] being set to RPD_A0 for all n; although subsequent uses of RPD_A0[n] can override this for selected n.

RPD_A1[m]
: α_1 [m^{-1}] of cable m ($1 \leq m \leq$ N_RPD) at the reference frequency RPD_FREF[m]. This is an additional parameter included to improve accuracy, but is not implemented in the Memo 170 model as of Version 3. Set to 0.0 unless otherwise specified. RPD_A1 (without [m]) will result in RPD_A1[n] being set to RPD_A1 for all n; although subsequent uses of RPD_A1[n] can override this for selected n.

RPD_FREF[m]
: Frequency [Hz] at which the parameters RPD_A0[m] and RPD_A1[m] are specified. Set to 10.0e+6 (10 MHz) unless otherwise specified. RPD_FREF (without [m]) will result in RPD_FREF[n] being set to RPD_FREF for all n; although subsequent uses of RPD_FREF[n] can override this for selected n.

RPD_STR[m]
: Coefficient of stretching [unitless] for cable m ($1 \leq m \leq$ N_RPD). RPD_LENG[m] is multiplied by this prior to computation of cable gain or delay. Set to 1.0 unless otherwise specified. RPD_STR (without [m]) will result in RPD_STR[n] being set to RPD_STR for all n; although subsequent uses of RPD_STR[n] can override this for selected n.

RPD_ANT[m]
: Antenna to which cable m ($1 \leq m \leq$ N_RPD) is ultimately connected. Normally in the range 1 to $2 \times$ N_STD. A negative value means the cable is connected at its input, but not at its output. A value of 0 means this cable is disconnected at both ends, or that its connections are unknown. Will be set to m unless otherwise specified.

2.6 SEP Parameters

Note that a "SEP port" is defined as the path from the jack on the outside of the shelter, to the end of the cable that connects to the ASP input.

N_SEP
: Maximum number of ports through SEP; expected to be ≤ 512.

SEP_ID[m]
: Label which unambiguously identifies SEP port m ($1 \leq m \leq$ N_SEP) on the SEP panel. Set

to "UNK" unless otherwise specified.

SEP_STAT[m]
: The status of SEP port m ($1 \leq m \leq$ N_SEP). See Note 1. Will be set to 3 ("OK") unless otherwise specified.

SEP_CABL[m]
: Label or tag which unambiguously identifies the cable that connects the SEP panel to the ASP input. Set to "UNK" unless otherwise specified.

SEP_LENG[m]
: Length [m] of the cable that connects the SEP panel to the ASP input. Will be set to 0 unless otherwise specified. SEP_LENG (without [m]) will result in SEP_LENG[n] being set to SEP_LENG for all n; although subsequent uses of SEP_LENG[n] can override this for selected n.

SEP_DESI[m]
: An integer code which identifies the design of SEP port m ($1 \leq m \leq$ N_SEP), including the cable to ASP. See Note 2. Design information expected to be indexed by this code includes cable type, electrical and mechanical descriptions, frequency-domain transfer function described as coefficients in a polynomial fit (representative of all cables with this design code). Will be set to "1" unless otherwise specified. Use "0" to indicate that design is unknown or undocumented. SEP_DESI (without [m]) will result in SEP_DESI[n] being set to SEP_DESI for all n; although subsequent uses of SEP_DESI[n] can override this for selected n.

SEP_GAIN[m]
: Gain [dB] of SEP port m ($1 \leq m \leq$ N_SEP) including the cable to ASP, at the reference frequency of 38 MHz. Will be set to 0 unless otherwise specified. SEP_GAIN (without [m]) will result in SEP_GAIN[n] being set to SEP_GAIN for all n; although subsequent uses of SEP_GAIN[n] can override this for selected n.

SEP_ANT[m]
: Antenna to which SEP port m ($1 \leq m \leq$ N_SEP) is ultimately connected. Normally in the range 1 to $2 \times$ N_STD. A negative value means the SEP port is connected at its input, but not at its output. A value of 0 means this SEP port is disconnected at both ends, or that its connections are unknown. Will be set to m unless otherwise specified.

2.7 ARX Board Parameters

N_ARB
: Maximum number of ARX boards; expected to be ≤ 32.

N_ARBCH
: Maximum number of channels per ARX board; expected to be ≤ 16.

ARB_ID[m]
: Label or serial number which unambiguously identifies ARX board m ($1 \leq m \leq$ N_ARB). Maximum 10 characters. Will be set to "UNK" unless otherwise specified.

ARB_SLOT[m]
: Unambiguous identification of the slot of the ASP chassis in which ARX board m ($1 \leq m \leq$ N_ARB) is installed. Will be set to 0 unless otherwise specified.

ARB_DESI[m]
: An integer code which identifies the design of ARX board m ($1 \leq m \leq$ N_ARB). See Note 2. Design information expected to be indexed by this code includes board revision number, electrical and/or mechanical descriptions, frequency-domain transfer function described as coefficients in a polynomial fit (representative of all ARX board channels with this design code). Will be set to "1" unless otherwise specified. Use "0" to indicate that design is unknown or undocumented. ARB_DESI (without [m]) will result in ARB_DESI[n] being set to ARB_DESI for all n; although subsequent uses of ARB_DESI[n] can override this for selected n.

2.7.1 ARX Board Power Sources

ASP power sources must be identified in order of ARB index m, with each ARB's source being specified using ARB_RACK[m] and ARB_PORT[m] keywords, in that order.

ARB_RACK[m]
: From the perspective of SHL, this is the rack (1 to N_PWR_RACK) in which the power supply powering this ARX board is located. A value of 0 means this parameter is unknown. This parameter is used in conjunction with ARB_PORT[m] to identify the power source for this ARX board. Will be set to 0 unless otherwise specified.

ARB_PORT[m]
: From the perspective of SHL, this is the power port corresponding to the power supply powering this ARX board. A value of 0 means this parameter is unknown. This parameter is used in conjunction with ARB_RACK[m] to identify the power source for this ARX board. Will be set to 0 unless otherwise specified.

2.7.2 ARX Board Channel Parameters

Indexed by board m and channel p, where $1 \leq p \leq$ N_ARBCH.

ARB_STAT[m][p]
: The status of channel p ($1 \leq p \leq$ N_ARBCH) of ARX board m ($1 \leq m \leq$ N_ARB). See Note 1. This will be set to 3 ("OK") unless otherwise specified.

ARB_GAIN[m][p]
: Maximum gain [dB] of channel p ($1 \leq p \leq$ N_ARBCH) of ARX board m ($1 \leq m \leq$ N_ARB), at the reference frequency of 38 MHz in full-bandwidth mode. "Maximum gain" means gain when programmable attenuation is minimum. Will be set to 67.0 unless otherwise specified. ARB_GAIN (without [m][p]) will result in ARB_GAIN[m][p] being set to ARB_GAIN for all m and p; although subsequent uses of ARB_GAIN[m][p] can override this for the selected m and p.

ARB_ANT[m][p]
: Antenna that channel p ($1 \leq p \leq$ N_ARBCH) of ARX board m ($1 \leq m \leq$ N_ARB) is ultimately connected to. A negative value means the channel is connected at its input, but not at its output. A value of 0 means this channel is disconnected at both ends, or that its connections are unknown; this is the default if not specified.

ARB_IN[m][p]
: Label unambiguously identifying the input connector to channel p ($1 \leq p \leq$ N_ARBCH) of ARX

board m ($1 \leq m \leq$ N_ARB) on the ASP rack. Maximum 10 characters. Will be set to "UNK" unless otherwise specified.

ARB_OUT[m][p]
: Label unambiguously identifying the output connector from channel p ($1 \leq p \leq$ N_ARBCH) of ARX board m ($1 \leq m \leq$ N_ARB) on the ASP rack. Maximum 10 characters. Will be set to "UNK" unless otherwise specified.

2.8 SNAP Board Parameters

Although these parameters use the "SNAP" naming convention, they apply to any digitizer board used by NDP, including ZCU102-based systems. The key difference is the number of channels per board: set N_SNAPCH to match the hardware in use (e.g., 64 for SNAP boards, or 32 for ZCU102 boards).

N_SNAP
: Number of digitizer boards; expected to be ≤ 16.

N_SNAPCH
: Number of channels per board; expected to be ≤ 64. Set to match the hardware in use.

SNAP_ID[m]
: Label or serial number which unambiguously identifies digitizer board m ($1 \leq m \leq$ N_SNAP). Maximum 10 characters. Will be set to "UNK" unless otherwise specified.

SNAP_SLOT[m]
: Unambiguous identification of the slot in which digitizer board m ($1 \leq m \leq$ N_SNAP) is installed. Maximum 10 characters. Will be set to "UNK" unless otherwise specified.

SNAP_DESI[m]
: An integer code which identifies the design of digitizer board m ($1 \leq m \leq$ N_SNAP). See Note 2. Design information expected to be indexed by this code includes board revision number, firmware version, and bandpass described as coefficients in a polynomial fit. Will be set to "1" unless otherwise specified. Use "0" to indicate unknown/undocumented design.

2.8.1 SNAP Board Channel Parameters

Indexed by board m and channel p, where $1 \leq p \leq$ N_SNAPCH.

SNAP_STAT[m][p]
: The status of channel p ($1 \leq p \leq$ N_SNAPCH) of digitizer board m ($1 \leq m \leq$ N_SNAP). See Note 1. This will be set to 3 ("OK") unless otherwise specified.

SNAP_INR[m][p]
: Label unambiguously identifying the *rack* input connector for channel p ($1 \leq p \leq$ N_SNAPCH) of digitizer board m ($1 \leq m \leq$ N_SNAP) on the NDP rack. Maximum 10 characters. Will be set to "UNK" unless otherwise specified.

SNAP_INC[m][p]
: Label unambiguously identifying the *chassis* (i.e., inside the rack) input connector for channel p ($1 \leq p \leq$ N_SNAPCH) of digitizer board m ($1 \leq m \leq$ N_SNAP) on the NDP rack. Maximum 10 characters. Will be set to "UNK" unless otherwise specified.

`SNAP_ANT[`m`][`p`]`
: Antenna that channel p ($1 \leq p \leq$ `N_SNAPCH`) of digitizer board m ($1 \leq m \leq$ `N_SNAP`) is ultimately connected to. A value of 0 means this channel is not connected, or that its connection is unknown. If not specified otherwise, will be set to 0.

2.9 GPU Server Parameters

`N_SERVER`
: Number of GPU servers; expected to be ≤ 5.

`SERVER_ID[`m`]`
: Label or serial number (max 10 characters), $1 \leq m \leq$ `N_SERVER`.

`SERVER_SLOT[`m`]`
: Slot designation (max 10 characters).

`SERVER_STAT[`m`]`
: Status. See Note 1.

`SERVER_DESI[`m`]`
: Design code. See Note 2.

2.10 MCS-DR Parameters

`N_DR`
: Number of data recorders; expected to be ≤ 5.

`DR_STAT[`m`]`
: The status of MCS-DR m ($1 \leq m \leq$ `N_DR`). See Note 1. Will be set to 3 ("OK") unless otherwise specified.

`DR_ID[`m`]`
: Serial number which unambiguously identifies MCS-DR m ($1 \leq m \leq$ `N_DR`). Maximum 10 characters. Will be set to `"UNK"` unless otherwise specified.

`DR_PC[`m`]`
: The model of this MCS-DR PC. Will be set to `"UNK"` unless otherwise specified. *Values currently in use are "XPS435" and "T1500".*

`DR_NDP[`m`]`
: Which NDP output this MCS-DR is connected to. Values are 1–4 for beam outputs, and 5 for TBT/TBS. Will be set to 0 (not connected) unless otherwise specified.

2.11 Power Parameters

`N_PWR_RACK`
: Maximum number of racks, from the perspective of SHL; expected to be ≤ 8.

`N_PWR_PORT[`m`]`
: Maximum number of power ports in rack m ($1 \leq m \leq$ `N_PWR_RACK`), from the perspective of SHL; expected to be ≤ 50. Will be set to 0 (no ports) unless otherwise specified.

`PWR_SS[`m`][`p`]`
: Subsystem that receives power from port p ($1 \leq p \leq$ `N_PWR_PORT`) of rack m ($1 \leq m \leq$

N_PWR_RACK). Valid values: SHL, ASP, NDP, MCS, DR1–DR5. A value of "UNK" means this port is not connected, or that its connection is unknown.

PWR_NAME[m][p]
: Specific item that receives power from port p ($1 \leq p \leq$ N_PWR_PORT) of rack m ($1 \leq m \leq$ N_PWR_RACK). A value of "UNK" means this port is not connected, or that its connection is unknown. Valid values depend on PWR_SS:

  - For PWR_SS[m][p] = SHL, valid values are MCS, others TBD.
  - For PWR_SS[m][p] = ASP, valid values are MCS, FEE, ARX, FAN.
  - For PWR_SS[m][p] = NDP, valid values are MCS, FPG (FPGA boards), SVR (servers), FAN, SYN (synthesizer module), and SWI (10GbE switch).
  - For PWR_SS[m][p] = MCS, valid values are SCH (Scheduler), EXE (Executive), TP (Task Processor), CH (Command Hub), and GW (Gateway).
  - For PWR_SS[m][p] = DR1–DR5, valid values are PC, DS1 (DRSU 1), and DS2 (DRSU 2).

  It should be noted that while this information is largely (but not exactly) redundant with respect to the _RACK and _PORT parameters for subsystems, the former is intended primarily as an aid to operators and maintainers. MCS may use either for actionable control decisions, so it is important that they be consistent.

2.12 Array Pointing Correction Parameters

PC_AXIS_TH
: Array pointing correction axis rotation in the elevation (theta) plane, in degrees.

PC_AXIS_PH
: Array pointing correction axis rotation in the azimuth (phi) plane, in degrees.

PC_ROT
: Array pointing correction rotation, in degrees.

2.13 Station-Wide Configuration Parameters

MCS_CRA
: "Configuration request authority" policy to be used by MCS when processing requests to set FEE and ASP parameters (which obviously apply station-wide) in session definition files. "0" means that MCS sets FEE and ASP parameters according to the information in the SSMIF, and any requests for changes are ignored. "1" means that the FEE and ASP parameters set by the SSMIF are treated as defaults, and that a session may be able to change them. See the discussion of the SESSION_CRA keyword in the MCS Observing document for additional details.

MRP_*sss*
: This sets the station default recording period for the MIB associated with the level-1 subsystem *sss*, where *sss* is the usual three-letter acronym (e.g., "ASP", "NDP", etc.). Integer minutes. For example: 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" (default). Note that

the setting of this parameter does not imply anything about how often the MIB is *updated*; see `MUP_sss`. Typically, `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. Note also that it is possible for observation sessions to temporarily override these settings depending on `MCS_CRA` and the session definition.

`MUP_sss`

This sets the station default update period for the MIB associated with the level-1 subsystem `sss`, where `sss` is the usual three-letter acronym (e.g., "ASP", "NDP", etc.). Integer minutes. For example: `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)" (default). It should be noted that there is only one set of MIBs for the station, and that they are common to all sessions.

When invoked, the order of invocation of subsystems must be: ASP, NDP, DR1, DR2, DR3, DR4, DR5, SHL, MCS. Note also that it is possible for observation sessions to temporarily override these settings depending on `MCS_CRA` and the session definition.

`FEE[`n`]`

Controls power for the FEE on stand n. `"1"` = "on", `"0"` = "off". $1 \leq n \leq 256$. `FEE` (without `[`n`]`) will result in `FEE[`n`]` being set to `FEE` for all n; although subsequent uses of `FEE[`n`]` can override this for the selected n. Otherwise, must be listed in order of increasing n. Note also that it is possible for observation sessions to temporarily override these settings depending on `MCS_CRA` and the session definition.

`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" (default), `"2"` = "reduced", and `"3"` = "off". $1 \leq n \leq 256$. `ASP_FLT` (without `[`n`]`) will result in `ASP_FLT[`n`]` being set to `ASP_FLT` for all n; although subsequent uses of `ASP_FLT[`n`]` can override this for the selected n. Otherwise, must be listed in order of increasing n. Note also that it is possible for observation sessions to temporarily override these settings depending on `MCS_CRA` and the session definition.

`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 (default) and 15. $1 \leq n \leq 256$. `ASP_AT1` (without `[`n`]`) will result in `ASP_AT1[`n`]` being set to `ASP_AT1` for all n; although subsequent uses of `ASP_AT1[`n`]` can override this for the selected n. Otherwise, must be listed in order of increasing n. Note also that it is possible for observation sessions to temporarily override these settings depending on `MCS_CRA` and the session definition.

`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 (default) and 15. $1 \leq n \leq 256$. `ASP_AT2` (without `[`n`]`) will result in `ASP_AT2[`n`]` being set to `ASP_AT2` for all n; although subsequent uses of `ASP_AT2[`n`]` can override this for the selected n. Otherwise, must be listed in order of increasing n. Note also that it is possible for observation sessions to temporarily override these settings depending on `MCS_CRA` and the session definition.

`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 (default) and 31. $1 \leq n \leq 256$. `ASP_AT3` (without `[`n`]`) will result in `ASP_AT3[`n`]` being set to `ASP_AT3` for all n; although subsequent uses of `ASP_AT3[`n`]` can override this for the selected n. Otherwise, must be listed in order of increasing n. Note also that it is possible for observation sessions to temporarily override these settings depending on `MCS_CRA` and the session definition.

`DRX_GAIN`
: This corresponds to the NDP DRX command parameter "DRX_GAIN". This is an integer value between 0 (default) and 15. Note it is possible for observation sessions to temporarily override these settings depending on `MCS_CRA` and the session definition.

2.14 Notes

Note 1: For status (`_STAT`) entries, the valid codes are:

- 3 = "OK"
- 2 = "Suspect; possibly bad (If used, provide warning)"
- 1 = "Bad (Don't use)"
- 0 = "Not Installed"

Note 2: The details of the use of `_DESI` parameters has not yet been worked out.

3 Format of the Station Dynamic MIB

The station dynamic MIB (SDM) is a file, typically named `sdm.dat`. It can be read and interpreted using the MCS/Task Processor utility `tprs`. The format is defined as a C-language structure:

```
#include <sys/time.h>

/* subsystem status */
struct subsystem_status_struct {
  int summary; /* SUMMARY; one of LWA_SIDSUM_* */
  char info[256]; /* INFO */
  struct timeval tv; /* time SUMMARY and INFO were last updated */
  };

/* sub-sub-system status */
/* note: this is the subset of the SSMIF with things that can change */
struct subsubsystem_status_struct {
  int eFEEStat[ME_MAX_NFEE]; /* FEE_STAT[] */
  int eRPDStat[ME_MAX_NRPD]; /* RPD_STAT[] */
  int eSEPStat[ME_MAX_NSEP]; /* SEP_STAT[] */
  int eARBStat[ME_MAX_NARB][ME_MAX_NARBCH]; /* ARB_STAT[][] */
  int eSnapStat[ME_MAX_NSNAP][ME_MAX_NSNAPCH]; /* SNAP_STAT[][] */
  int eServerStat[ME_MAX_NSERVER]; /* SERVER_STAT[] */
  int eDRStat[ME_MAX_NDR]; /* DR_STAT[] */
  };

/* this sub-structure is used in both the ssmif and sdm */
```

```
struct station_settings_struct {
  signed short int mrp_asp; // SESSION_MRP_ASP // MRP_ASP
  signed short int mrp_ndp; // SESSION_MRP_NDP // MRP_NDP
  signed short int mrp_dr1; // SESSION_MRP_DR1 // MRP_DR1
  signed short int mrp_dr2; // SESSION_MRP_DR2 // MRP_DR2
  signed short int mrp_dr3; // SESSION_MRP_DR3 // MRP_DR3
  signed short int mrp_dr4; // SESSION_MRP_DR4 // MRP_DR4
  signed short int mrp_dr5; // SESSION_MRP_DR5 // MRP_DR5
  signed short int mrp_shl; // SESSION_MRP_SHL // MRP_SHL
  signed short int mrp_mcs; // SESSION_MRP_MCS // MRP_MCS
  signed short int mup_asp; // SESSION_MUP_ASP // MUP_ASP
  signed short int mup_ndp; // SESSION_MUP_NDP // MUP_NDP
  signed short int mup_dr1; // SESSION_MUP_DR1 // MUP_DR1
  signed short int mup_dr2; // SESSION_MUP_DR2 // MUP_DR2
  signed short int mup_dr3; // SESSION_MUP_DR3 // MUP_DR3
  signed short int mup_dr4; // SESSION_MUP_DR4 // MUP_DR4
  signed short int mup_dr5; // SESSION_MUP_DR5 // MUP_DR5
  signed short int mup_shl; // SESSION_MUP_SHL // MUP_SHL
  signed short int mup_mcs; // SESSION_MUP_MCS // MUP_MCS
  signed short int fee[LWA_MAX_NSTD]; // OBS_FEE[LWA_MAX_NSTD][2]
  signed short int asp_flt[LWA_MAX_NSTD]; // OBS_ASP_FLT[LWA_MAX_NSTD]
  signed short int asp_at1[LWA_MAX_NSTD]; // OBS_ASP_AT1[LWA_MAX_NSTD]
  signed short int asp_at2[LWA_MAX_NSTD]; // OBS_ASP_AT2[LWA_MAX_NSTD]
  signed short int asp_at3[LWA_MAX_NSTD]; // OBS_ASP_AT3[LWA_MAX_NSTD]
  signed short int drx_gain; // OBS_DRX_GAIN // DRX_GAIN
  };

/* station dynamic MIB (SDM) */
struct sdm_struct {
  struct subsystem_status_struct station; /* Station overall status */
  struct subsystem_status_struct shl; /* SHL status */
  struct subsystem_status_struct asp; /* ASP status */
  struct subsystem_status_struct ndp; /* NDP status */
  struct subsystem_status_struct dr[ME_MAX_NDR]; /* DR# status (0=DR1,1=DR2,...) */
  struct subsubsystem_status_struct ssss; /* correspond to SSMIF "stat" items */
  int ant_stat[ME_MAX_NSTD][2]; /* corresponds to sc.Stand[i].Ant[k].iSS,
                                   but dynamically updated */
  int ndpo_stat[ME_MAX_NDR]; /* corresponds to sc.NDPO[i].iStat,
                                   but dynamically updated */
  struct station_settings_struct settings; /* these are the current,
                                              dynamically-varying settings */
  };

struct sdm_struct sdm; /* so finally this is the sdm */
```

In the above, the `int` type is a 4-byte little-endian integer and the `short int` type is a 2-byte little-endian integer.

The fields are interpreted as follows:

`summary`
: Maps to MCS Common ICD MIB entry 1.1 (`SUMMARY`). Valid values:

  - 0 (`LWA_SIDSUM_NULL`) — Not normally used.

- 1 (`LWA_SIDSUM_NORMAL`) — Normal.
- 2 (`LWA_SIDSUM_WARNING`) — Warning.
- 3 (`LWA_SIDSUM_ERROR`) — Error.
- 4 (`LWA_SIDSUM_BOOTING`) — Booting.
- 5 (`LWA_SIDSUM_SHUTDWN`) — Shutdown.
- 6 (`LWA_SIDSUM_UNK`) — Status unknown.

`info`
: Human-readable text explaining `summary`.

`tv`
: Time at which `SUMMARY` and `INFO` were last updated, represented as the Linux/C `timeval` type [2].

`e...Stat[]`
: Status codes for individual components. The value can only be less than or equal to the corresponding SSMIF value. Status codes:

  - 0 = Not installed.
  - 1 = Bad.
  - 2 = Suspect.
  - 3 = OK.

`ant_stat[][]`
: Status for the entire path from stand/antenna through SNAP.

`ndpo_stat[]`
: Status for the path from server to the associated DR.

`settings`
: Correspond to comparably-named items in MCS0030 and always reflect the current state. A value of -1 indicates unknown.

A Example of a Station Static MIB Initialization File

The following is an example of a Station Static MIB Initialization File for a small station with 16 stands. It is based on a subset of the LWA1 SSMIF and illustrates the use of default values. Where defaults are sufficient (e.g., antenna-to-stand mapping, orientations, cable parameters), only comments indicating the use of defaults are shown rather than enumerating every value.

```
# ------------------------------------------------
# --- Station Static MIB Initialization File ---
# ------------------------------------------------
# Example SSMIF for documentation purposes.
# Based on a subset (16 stands) of the LWA1 SSMIF.

FORMAT_VERSION 10
STATION_ID VL
GEO_N +34.068894 # N Lat [deg]
GEO_E -107.628350 # E Lon [deg]
```

```
GEO_EL 2133.6 # [m] above MSL
N_STD 16

# ----------------------------------------
# --- Stand Positions --------------------
# ----------------------------------------
STD_LX[1] -1.006
STD_LY[1] -54.590
STD_LZ[1] +3.204
STD_LX[2] +0.232
STD_LY[2] -49.301
STD_LZ[2] +3.075
STD_LX[3] -2.621
STD_LY[3] -43.460
STD_LZ[3] +2.802
STD_LX[4] +1.350
STD_LY[4] -39.405
STD_LZ[4] +2.707
STD_LX[5] -3.380
STD_LY[5] -31.576
STD_LZ[5] +2.503
STD_LX[6] +1.717
STD_LY[6] -26.046
STD_LZ[6] +2.298
STD_LX[7] -0.060
STD_LY[7] -16.578
STD_LZ[7] +2.020
STD_LX[8] +0.316
STD_LY[8] -10.798
STD_LZ[8] +1.817
STD_LX[9] -0.787
STD_LY[9] -4.370
STD_LZ[9] +1.683
STD_LX[10] +1.893
STD_LY[10] +0.863
STD_LZ[10] +1.536
STD_LX[11] -0.641
STD_LY[11] +5.666
STD_LZ[11] +1.403
STD_LX[12] -1.245
STD_LY[12] +14.437
STD_LZ[12] +1.052
STD_LX[13] -0.813
STD_LY[13] +19.825
STD_LZ[13] +0.893
STD_LX[14] -0.055
STD_LY[14] +25.202
STD_LZ[14] +0.683
STD_LX[15] +1.960
STD_LY[15] +29.739
STD_LZ[15] +0.511
STD_LX[16] +2.872
STD_LY[16] +35.432
STD_LZ[16] +0.284

# ------------------------------------
# --- Antenna # -> Stand # mapping ---
# ------------------------------------
# Defaults are used: antennas 1 & 2 on stand 1, 3 & 4 on stand 2, etc.
```

```
# ----------------------------
# --- Antenna orientations ---
# ----------------------------
# Defaults are used: odd antennas = 0 (N-S), even antennas = 1 (E-W).

# ----------------------
# --- Antenna Status ---
# ----------------------
# Defaults are used (3 = OK) except:
ANT_STAT[9] 1 # Bad

# -------------------------------
# --- Antenna mis-orientation ---
# -------------------------------
# Defaults are used (0.0) except:
ANT_PHI[7] +15.0
ANT_PHI[8] +15.0

# ---------------------------------
# --- Antenna design information ---
# ---------------------------------
# Default is used (ANT_DESI = 1).

# ----------------------
# --- FEEs -------------
# ----------------------
N_FEE 16

# FEE_ID, FEE_STAT, FEE_DESI, FEE_GAI1, FEE_GAI2 all use defaults.
# FEE_ANT1/FEE_ANT2 mapping uses defaults.

# ---------------------------------------
# --- FEE Power Source Identification ----
# ---------------------------------------
FEE_RACK[1] 1
FEE_PORT[1] 1
FEE_RACK[2] 1
FEE_PORT[2] 1
FEE_RACK[3] 1
FEE_PORT[3] 1
FEE_RACK[4] 1
FEE_PORT[4] 1
FEE_RACK[5] 1
FEE_PORT[5] 1
FEE_RACK[6] 1
FEE_PORT[6] 1
FEE_RACK[7] 1
FEE_PORT[7] 1
FEE_RACK[8] 1
FEE_PORT[8] 1
FEE_RACK[9] 1
FEE_PORT[9] 1
FEE_RACK[10] 1
FEE_PORT[10] 1
FEE_RACK[11] 1
FEE_PORT[11] 1
FEE_RACK[12] 1
FEE_PORT[12] 1
```

```
FEE_RACK[13] 1
FEE_PORT[13] 1
FEE_RACK[14] 1
FEE_PORT[14] 1
FEE_RACK[15] 1
FEE_PORT[15] 1
FEE_RACK[16] 1
FEE_PORT[16] 1

# ----------------------
# --- RPD (Cables) -----
# ----------------------
N_RPD 32

RPD_ID[1] EXK-001-138 (Gray)
RPD_ID[2] EXK-001-138 (Black)
RPD_ID[3] EXK-002-133 (Gray)
RPD_ID[4] EXK-002-133 (Black)
RPD_ID[5] EXK-003-134 (Gray)
RPD_ID[6] EXK-003-134 (Black)
RPD_ID[7] EXK-004-126 (Gray)
RPD_ID[8] EXK-004-126 (Black)
RPD_ID[9] EXK-005-117 (Gray)
RPD_ID[10] EXK-005-117 (Black)
RPD_ID[11] EXK-006-121 (Gray)
RPD_ID[12] EXK-006-121 (Black)
RPD_ID[13] EXK-007-105 (Gray)
RPD_ID[14] EXK-007-105 (Black)
RPD_ID[15] EXK-008-100 (Gray)
RPD_ID[16] EXK-008-100 (Black)
RPD_ID[17] EXK-009-091 (Gray)
RPD_ID[18] EXK-009-091 (Black)
RPD_ID[19] EXK-010-086 (Gray)
RPD_ID[20] EXK-010-086 (Black)
RPD_ID[21] EXK-011-096 (Gray)
RPD_ID[22] EXK-011-096 (Black)
RPD_ID[23] EXK-012-089 (Gray)
RPD_ID[24] EXK-012-089 (Black)
RPD_ID[25] EXK-013-083 (Gray)
RPD_ID[26] EXK-013-083 (Black)
RPD_ID[27] EXK-014-081 (Gray)
RPD_ID[28] EXK-014-081 (Black)
RPD_ID[29] EXK-015-082 (Gray)
RPD_ID[30] EXK-015-082 (Black)
RPD_ID[31] EXK-016-077 (Gray)
RPD_ID[32] EXK-016-077 (Black)

# RPD_STAT, RPD_DESI use defaults (3, 1).

RPD_LENG[1] 138.00
RPD_LENG[2] 138.00
RPD_LENG[3] 133.00
RPD_LENG[4] 133.00
RPD_LENG[5] 134.00
RPD_LENG[6] 134.00
RPD_LENG[7] 126.00
RPD_LENG[8] 126.00
RPD_LENG[9] 117.00
RPD_LENG[10] 117.00
```

```
RPD_LENG[11] 121.00
RPD_LENG[12] 121.00
RPD_LENG[13] 105.00
RPD_LENG[14] 105.00
RPD_LENG[15] 100.00
RPD_LENG[16] 100.00
RPD_LENG[17] 91.00
RPD_LENG[18] 91.00
RPD_LENG[19] 86.00
RPD_LENG[20] 86.00
RPD_LENG[21] 96.00
RPD_LENG[22] 96.00
RPD_LENG[23] 89.00
RPD_LENG[24] 89.00
RPD_LENG[25] 83.00
RPD_LENG[26] 83.00
RPD_LENG[27] 81.00
RPD_LENG[28] 81.00
RPD_LENG[29] 82.00
RPD_LENG[30] 82.00
RPD_LENG[31] 77.00
RPD_LENG[32] 77.00

# RPD_VF, RPD_DD, RPD_A0, RPD_A1, RPD_FREF, RPD_STR defaults:
# (per-cable RPD_VF/DD/A0/A1/FREF use defaults; only RPD_STR overridden)

RPD_STR[1] 1.0085
RPD_STR[2] 0.9955
RPD_STR[3] 1.0070
RPD_STR[4] 0.9900
RPD_STR[5] 1.0040
RPD_STR[6] 0.9925
RPD_STR[7] 1.0200
RPD_STR[8] 1.0040
RPD_STR[9] 1.0470
RPD_STR[10] 1.0100
RPD_STR[11] 1.0240
RPD_STR[12] 1.0145
RPD_STR[13] 1.0180
RPD_STR[14] 1.0150
RPD_STR[15] 1.0510
RPD_STR[16] 1.0170
RPD_STR[17] 1.0030
RPD_STR[18] 1.0095
RPD_STR[19] 1.0210
RPD_STR[20] 1.0195
RPD_STR[21] 1.0060
RPD_STR[22] 0.9810
RPD_STR[23] 1.0635
RPD_STR[24] 1.0595
RPD_STR[25] 1.0085
RPD_STR[26] 1.0475
RPD_STR[27] 1.0360
RPD_STR[28] 1.0160
RPD_STR[29] 1.0255
RPD_STR[30] 1.0120
RPD_STR[31] 1.0240
RPD_STR[32] 1.0175
```

```
# RPD_ANT uses defaults (RPD_ANT[n] = n).

# ----------------------
# --- SEP --------------
# ----------------------
N_SEP 32

# SEP_ID, SEP_STAT use defaults.

SEP_CABL[1] 001-NS
SEP_CABL[2] 001-EW
SEP_CABL[3] 002-NS
SEP_CABL[4] 002-EW
SEP_CABL[5] 003-NS
SEP_CABL[6] 003-EW
SEP_CABL[7] 004-NS
SEP_CABL[8] 004-EW
SEP_CABL[9] 005-NS
SEP_CABL[10] 005-EW
SEP_CABL[11] 006-NS
SEP_CABL[12] 006-EW
SEP_CABL[13] 007-NS
SEP_CABL[14] 007-EW
SEP_CABL[15] 008-NS
SEP_CABL[16] 008-EW
SEP_CABL[17] 009-NS
SEP_CABL[18] 009-EW
SEP_CABL[19] 010-NS
SEP_CABL[20] 010-EW
SEP_CABL[21] 011-NS
SEP_CABL[22] 011-EW
SEP_CABL[23] 012-NS
SEP_CABL[24] 012-EW
SEP_CABL[25] 013-NS
SEP_CABL[26] 013-EW
SEP_CABL[27] 014-NS
SEP_CABL[28] 014-EW
SEP_CABL[29] 015-NS
SEP_CABL[30] 015-EW
SEP_CABL[31] 016-NS
SEP_CABL[32] 016-EW

# SEP_LENG, SEP_DESI, SEP_GAIN, SEP_ANT use defaults.

# ----------------------
# --- ASP (ARX) --------
# ----------------------
N_ARB 2
N_ARBCH 16

ARB_ID[1] 0343
ARB_ID[2] 0335

ARB_SLOT[1] 11
ARB_SLOT[2] 12

ARB_DESI[1] 5
ARB_DESI[2] 5
```

```
ARB_RACK[1] 1
ARB_PORT[1] 2
ARB_RACK[2] 1
ARB_PORT[2] 2

# ARB_STAT, ARB_GAIN use defaults (3, 67.0).
# ARB_ANT uses defaults (sequential mapping).

ARB_IN[1][1] 4_1
ARB_IN[1][2] 4_2
ARB_IN[1][3] 4_3
ARB_IN[1][4] 4_4
ARB_IN[1][5] 4_5
ARB_IN[1][6] 4_6
ARB_IN[1][7] 4_7
ARB_IN[1][8] 4_8
ARB_IN[1][9] 4_9
ARB_IN[1][10] 4_10
ARB_IN[1][11] 4_11
ARB_IN[1][12] 4_12
ARB_IN[1][13] 4_13
ARB_IN[1][14] 4_14
ARB_IN[1][15] 4_15
ARB_IN[1][16] 4_16
ARB_IN[2][1] 4_17
ARB_IN[2][2] 4_18
ARB_IN[2][3] 4_19
ARB_IN[2][4] 4_20
ARB_IN[2][5] 4_21
ARB_IN[2][6] 4_22
ARB_IN[2][7] 4_23
ARB_IN[2][8] 4_24
ARB_IN[2][9] 4_25
ARB_IN[2][10] 4_26
ARB_IN[2][11] 4_27
ARB_IN[2][12] 4_28
ARB_IN[2][13] 4_29
ARB_IN[2][14] 4_30
ARB_IN[2][15] 4_31
ARB_IN[2][16] 4_32

ARB_OUT[1][1] 1_1
ARB_OUT[1][2] 1_2
ARB_OUT[1][3] 1_3
ARB_OUT[1][4] 1_4
ARB_OUT[1][5] 1_5
ARB_OUT[1][6] 1_6
ARB_OUT[1][7] 1_7
ARB_OUT[1][8] 1_8
ARB_OUT[1][9] 1_9
ARB_OUT[1][10] 1_10
ARB_OUT[1][11] 1_11
ARB_OUT[1][12] 1_12
ARB_OUT[1][13] 1_13
ARB_OUT[1][14] 1_14
ARB_OUT[1][15] 1_15
ARB_OUT[1][16] 1_16
ARB_OUT[2][1] 1_17
ARB_OUT[2][2] 1_18
```

```
ARB_OUT[2][3] 1_19
ARB_OUT[2][4] 1_20
ARB_OUT[2][5] 1_21
ARB_OUT[2][6] 1_22
ARB_OUT[2][7] 1_23
ARB_OUT[2][8] 1_24
ARB_OUT[2][9] 1_25
ARB_OUT[2][10] 1_26
ARB_OUT[2][11] 1_27
ARB_OUT[2][12] 1_28
ARB_OUT[2][13] 1_29
ARB_OUT[2][14] 1_30
ARB_OUT[2][15] 1_31
ARB_OUT[2][16] 1_32

# ----------------------
# --- NDP (SNAP/ZCU) ---
# ----------------------
N_SNAP 1
N_SNAPCH 32

SNAP_ID[1] 0A351D7908

SNAP_SLOT[1] 1-1

# SNAP_DESI: 2 = SNAP2, 3 = ZCU102
SNAP_DESI[1] 3

# SNAP_STAT uses defaults (3 = OK).

SNAP_INR[1][1] 4_1
SNAP_INR[1][2] 4_2
SNAP_INR[1][3] 4_3
SNAP_INR[1][4] 4_4
SNAP_INR[1][5] 4_5
SNAP_INR[1][6] 4_6
SNAP_INR[1][7] 4_7
SNAP_INR[1][8] 4_8
SNAP_INR[1][9] 4_9
SNAP_INR[1][10] 4_10
SNAP_INR[1][11] 4_11
SNAP_INR[1][12] 4_12
SNAP_INR[1][13] 4_13
SNAP_INR[1][14] 4_14
SNAP_INR[1][15] 4_15
SNAP_INR[1][16] 4_16
SNAP_INR[1][17] 4_17
SNAP_INR[1][18] 4_18
SNAP_INR[1][19] 4_19
SNAP_INR[1][20] 4_20
SNAP_INR[1][21] 4_21
SNAP_INR[1][22] 4_22
SNAP_INR[1][23] 4_23
SNAP_INR[1][24] 4_24
SNAP_INR[1][25] 4_25
SNAP_INR[1][26] 4_26
SNAP_INR[1][27] 4_27
SNAP_INR[1][28] 4_28
SNAP_INR[1][29] 4_29
```

```
SNAP_INR[1][30] 4_30
SNAP_INR[1][31] 4_31
SNAP_INR[1][32] 4_32

# SNAP_INC uses defaults (UNK).
# SNAP_ANT uses defaults (sequential mapping).

# ----------------------
# --- GPU Servers ------
# ----------------------
N_SERVER 5

SERVER_ID[1] ndp
SERVER_ID[2] ndp1
SERVER_ID[3] ndp2
SERVER_ID[4] ndp3
SERVER_ID[5] ndp4

SERVER_SLOT[1] 1
SERVER_SLOT[2] 2
SERVER_SLOT[3] 3
SERVER_SLOT[4] 4
SERVER_SLOT[5] 5

# SERVER_STAT uses defaults (3 = OK).

# SERVER_DESI: 1 = head node, 2 = compute node
SERVER_DESI[1] 1
SERVER_DESI[2] 2
SERVER_DESI[3] 2
SERVER_DESI[4] 2
SERVER_DESI[5] 2

# ----------------------
# --- DR ---------------
# ----------------------
N_DR 5

DR_STAT[1] 3
DR_STAT[2] 3
DR_STAT[3] 3
DR_STAT[4] 3
DR_STAT[5] 3

DR_ID[1] DR1
DR_ID[2] DR2
DR_ID[3] DR3
DR_ID[4] DR4
DR_ID[5] DR5

DR_PC[1] X11SSW-F
DR_PC[2] X11SSW-F
DR_PC[3] X11SSW-F
DR_PC[4] X11SSW-F
DR_PC[5] X11SSW-F

# DR_NDP: 1-4 = beam outputs, 5 = TBT/TBS
DR_NDP[1] 1
DR_NDP[2] 2
```

```
DR_NDP[3] 3
DR_NDP[4] 4
DR_NDP[5] 5

# ----------------------
# --- Power ------------
# ----------------------
N_PWR_RACK 3

N_PWR_PORT[1] 4
N_PWR_PORT[2] 6
N_PWR_PORT[3] 6

PWR_SS[1][1] ASP
PWR_NAME[1][1] ARX

PWR_SS[1][2] ASP
PWR_NAME[1][2] FEE

PWR_SS[1][4] ASP
PWR_NAME[1][4] FAN

PWR_SS[2][1] NDP
PWR_NAME[2][1] FPG

PWR_SS[2][4] NDP
PWR_NAME[2][4] SYN

PWR_SS[2][5] NDP
PWR_NAME[2][5] FAN

PWR_SS[3][1] NDP
PWR_NAME[3][1] MCS

PWR_SS[3][3] NDP
PWR_NAME[3][3] SVR

PWR_SS[3][4] DR1
PWR_NAME[3][4] PC

PWR_SS[3][5] DR2
PWR_NAME[3][5] PC

PWR_SS[3][6] DR3
PWR_NAME[3][6] PC

# ----------------------
# --- Configuration ----
# ----------------------

MCS_CRA 1

DRX_GAIN 6

# ====================
# === STATUS CODES ===
# ====================
# 3 = "OK"
# 2 = "Suspect; possibly bad (If used, provide warning)"
```

```
# 1 = "Bad (Don't use)"
# 0 = "Not Installed"
```

References

1. S. Ellingson & J. Dowell, "LWA Station-Level Observing Procedure and Associated Metadata," Ver. 10, LWA Engineering Memo MCS0030, February 2026.

2. `http://linux.die.net/man/2/gettimeofday`