MCS Common ICD

Version 1.1

S. Ellingson, J. Dowell

February 2026

Long Wavelength Array Project

Change Record

Version	Date	Author(s)	Changes
0.2	2009-02-23	S. Ellingson	• Julian day (JDAY) changed to modified Julian day (MJD) • Fixed error in left- vs. right-justification • Pointed out that a UT day can have a leap second • More specific guidance for MIB labels
0.3	2009-03-20	S. Ellingson	• Fixed error in representation of MJD • Fixed error where SUBSYSTEM and SERIALNO were both index 1.4 • Settled on UDP over TCP • Maximum payload set to 4050 bytes • Added DATALEN field • MJD/MPM interpreted as send time • Added CKSUM field • Standardized response DATA field • Clarified multi-packet messages • Changed space notation in examples
0.4	2009-03-29	S. Ellingson	• INFO uses MIB labels not indices • Editing of message/packet/payload terms • Multipacket messages removed • Fixed RPT response examples • Guidance on MJD/MPM meanings • Standardized 3-second response time • Clarified invalid checksum handling • Edited SERIALNO field
1.0	2009-04-04	S. Ellingson	• Maximum message length increased to 8192 bytes • Removed CKSUM field
1.1	2026-02	J. Dowell	• Replaced DP with NDP throughout • Added unsolicited RPT response support • Updated organization address • Fixed MIB label length limit (40 → 32) to match implementation

Contents

1 Introduction and Scope

MCS stands for "monitoring and control system." As described in the LWA station architecture document [1], MCS monitors and controls ASP ("analog signal processing"), NDP ("Next Generation Digital Processor"), and other subsystems. The purpose of this interface control document (ICD) is to define a common interface between MCS and connected subsystems.

Compliance with this ICD is necessary, but not sufficient for integration with MCS. As explained below, this ICD provides a framework for the interface between MCS and subsystems which are directly connected to it, including electromechanical interfaces and protocol information. It is expected that subsystems connecting to MCS will cite this ICD in their subsystem-specific ICDs, and then specify subsystem-specific information as extensions within this framework.

This ICD does not apply to the interface between NDP and the data recording capability being provided by MCS, described in Section 3 of [2]. It also does not apply to the Heuristic Automation for LWA (HAL) or the SmartCopy systems.

2 Summary

1. The sole physical interface with MCS will be a single 1000BASE-T (full-duplex gigabit ethernet) connection over Category 6 ("Cat-6") cable terminated in RJ45 connectors.
2. The sole protocol interface with MCS will be UDP, with direct passing of messages using sockets.[1] IP addresses are static and defined in a separate document. Port assignments are defined in a separate document. The term "message" is defined henceforth to mean a single command or response, contained entirely within the data field of one or more UDP packets. A message will normally correspond to a single use of a "send()" or "recv()" function (with syntax dependent on the programming language, of course). Message structure is defined in Section 4.
3. The interface will operate according to a "polling" paradigm. Connected subsystems will never *initiate* communications, and will only respond to an MCS message to the extent required by the applicable ICD(s). Subsystems shall not communicate with subsystems other than MCS over this interface. The exception to this rule is that subsystems may send unsolicited `RPT` response messages to MCS; see Section 5.

3 MIB

"MIB" stands for "management information base."[2] The MIB provides a means for organizing subsystem status information that is jointly understood by communicating subsystems.

The MIB has an index/outline structure, as demonstrated by the MIB fragment in Table 2. (Note this fragment is an example only, shown only for the purposes of explaining the MIB concept.) In this fragment, each line is an "entry", consisting of an "index" (e.g., 2) and a "label" (e.g., A2). Each MIB index/label possibly also has an associated data value. A "branch" is a set of entries with a common index/label; for example, branch 2 (also known as "A2") contains the data values B21 =

[1] For the uninitiated, see `http://en.wikipedia.org/wiki/User_Datagram_Protocol` and/or `http://docs.python.org/library/socket.html`.

[2] The use of this term is a nod to the MIB concept used in the SNMP protocol, but the two MIBs are not the same, and in fact are different in many respects.

3.4, D221 = "PRR", and E222 = 7. Other examples: Branch 2.1 contains the data value B21 = 3.4 only, and branch 2.2 (also known as "C22") contains data values D221 = "PRR" and E222 = 7. Note that entries with "sub-entries" are for organizational purposes only (making it possible to refer to multiple entries using a single index/label), and do not contain data. For example, entries 2 and 2.2 have labels only and contain no data.

Table 2: A MIB fragment, provided as an example only.

Index	**Label**	**Data**	**Remarks**
2	`A2`		
2.1	`B21`	3.4	5 bytes, ASCII, base-10, decimal point allowed
2.2	`C22`		
2.2.1	`D221`	PRR	3 bytes, ASCII, alphanumeric
2.2.2	`E222`	7	2 bytes, ASCII, base-10 integer

MIB labels must consist only of letters (case is significant), integer numbers, and the underscore character. Spaces are not allowed. The length must be less than or equal to 32 characters.

Data referenced by MIB entries need not be ASCII, and can be raw binary. If raw binary, then the subsystem ICD must specify whether this is big- or little-endian. An example of the use of raw binary would be to represent the coefficients for a digital filter. The filter can be represented as a MIB branch where each entry is the raw bit values for one coefficient, or the entire filter can be represented as a single entry consisting of all coefficients concatenated into a contiguous sequence of bits.

Each subsystem communicating with MCS using this ICD must specify a MIB as part of a subsystem-specific ICD. This MIB consists of MCS-required MIB entries, plus additional MIB entries which are subsystem-specific. The MCS-required MIB entries are specified below.

1. `MCS-RESERVED`

   1.1. `SUMMARY` [7 bytes, ASCII/Alphanumeric]. Summary state of subsystem. Valid values are as follows:

   - `NORMAL`
   - `WARNING` (issue(s) found, but still fully operational)
   - `ERROR` (problems found which limit or prevent proper operation)
   - `BOOTING` (initializing system; not yet fully operational)
   - `SHUTDWN` (shutting down system; not ready for operation)

   1.2. `INFO` [maximum 256 bytes, ASCII]. When MIB entry 1.1 is `WARNING` or `ERROR`, this entry should begin with a list of MIB labels, separated by single spaces, and terminated by the character "!" (exclamation mark). The MIB labels should be those containing values indicating the problem condition. A human-readable text string which further explains `WARNING` and `ERROR` values may be included following the character "!". Any unused bytes at the end of the `INFO` string should be spaces. Use of this MIB entry when 1.1 is not `WARNING` or `ERROR` is subsystem-specific.

1.3. `LASTLOG` [maximum 256 bytes, ASCII]. Last internal log message. Human-readable text string, with format specified in the subsystem-specific ICD. A timestamp of some form should be included near the beginning of the string. Any unused bytes at the end of the string should be spaces.

1.4. `SUBSYSTEM` [3 bytes, ASCII/Alphanumeric]. 3 character string identifying the subsystem; e.g., "`NDP`", "`ASP`". All strings beginning with the characters "MC" are reserved. Other strings are assigned by the LWA Systems Engineer.[3]

1.5. `SERIALNO` [maximum 5 bytes, ASCII]. A string identifying the specific subsystem hardware "serial number". This string is assigned by the subsystem manufacturer in coordination with the LWA Systems Engineer.

1.6. `VERSION` [maximum 256 bytes, ASCII]. Version number of locally-installed software. May include additional information or elaboration; if so, the "principal" version number must appear first and be followed by a single space. Any unused bytes at the end of this string should be spaces.

1.{n}. *Additional MIB entries beginning "1." TBD*

{n}. *MIB entries numbered 2 or higher are subsystem-specific, and are defined in the associated ICDs*

4 Message Structure

Messages are contained entirely within the payload fields of UDP packets. The maximum size of a message is 8192 bytes. A message is subdivided into fields as defined below. All fields are required (except as indicated), contiguous, and must appear in the order indicated and with the number of bytes indicated. Unless indicated otherwise, data are right-justified in their fields, and padded with the character " " (space).

1. `DESTINATION` [3 bytes, ASCII/Alphanumeric]. This is the intended recipient of the message. Valid values are "`ALL`" (to be interpreted as "all subsystems receiving this message"), "`ASP`", "`NDP`", and "`MCS`". *(Other values will be added as necessary.)* Subsystems shall ignore any message not addressed to either the subsystem or "`ALL`".

2. `SENDER` [3 bytes, ASCII/Alphanumeric]. This is the subsystem sending the message. Valid values are the same as for `DESTINATION`.

3. `TYPE` [3 bytes, ASCII/Alphanumeric]. This field indicates the type of message. A list of message types is given in Section 5.

4. `REFERENCE` [9 bytes, ASCII/Numeric] (base-10 integer). MCS assigns reference numbers to messages. Reference numbers are assigned sequentially station-wide, so connected subsystems should not interpret gaps in the sequence as missed messages. Responses to MCS command messages use the same reference number appearing in the command message.

5. `DATALEN` [4 bytes, ASCII/Numeric] (base-10 integer). This is the number of bytes in the `DATA` field.

[3] One possible use of this entry is to facilitate subsystem discovery; e.g., MCS can send a `RPT` command message to "ALL" requesting MIB entry 1.4, and see who responds.

6. MJD [6 bytes, ASCII/Numeric] (base-10 integer). Integer part of the modified Julian day (MJD). For example: For Dec 28, 2008 UT this is "54828". See additional information below.
7. MPM [9 bytes, ASCII/Numeric] (base-10 integer). Milliseconds past UT midnight; see "MJD". (Note that there are 86,400,000 milliseconds in a UT day, except for days with a leap second.) See additional information below.
8. There is always a space following the MPM field.
9. DATA. [Variable length, variable format]. The contents of this field depend on the message TYPE; see Section 5.

The purpose of the MJD and MPM fields is primarily to confirm to the recipient that the sender has a consistent understanding of what time it is, and also to provide a convenient mechanism for keeping or searching logs. Subsystems may use this information to set local clocks, with the understanding that the accuracy of these times (due to non-deterministic OS- and transmission-related delays) is probably not better than a few milliseconds, and could be intermittently much worse. Unless explicitly indicated in associated subsystem ICDs, MJD and MPM should not be interpreted as the time at which the command is to take effect, nor should these be interpreted as being precisely the time at which the condition reported in a response was observed. If it is necessary to convey this information precisely, those times can alternatively be indicated separately as part of the DATA field. MJD/MPM should reflect the "best available estimate" of station time as known to the sender. A satisfactory "best available estimate" can be obtained simply by calling an appropriate time function immediately prior to assembling the message and sending it, and it is expected that this time will represent the time at which the message was actually transmitted to within a few milliseconds.

See Section 6 for examples of command and response messages.

5 Message Types

Messages from MCS are commands. Commands can request action, information, or both. Connected subsystems respond as specified by the message TYPE definition. The following is a list of message TYPEs that are common to all MCS interfaces. ICDs between MCS and specific subsystems may specify additional message TYPEs.

- "PNG" = Ping. The purposes of the command message are (1) to confirm that a commanded system is functioning, and (2) to disseminate or confirm time information. Upon receiving this message, the commanded subsystem (1) verifies that its local time is consistent with the time given in the received command message, updating if necessary; and (2) responds with a PNG response message. The DATA field of the command message is empty, the DATA field of the response message is limited to the standard response indicated below. See Section 6 for an example.
- "RPT" = Report. The purpose of this message TYPE is to provide a flexible method for reporting subsystem status. In the command message, the DATA field contains a label corresponding to MIB entry or branch, indicating that the commanded subsystem should respond with the current values of the MIB for that index/branch. The MIB data is provided as a contiguous block of data, with no delimiters or terminators (this is to avoid difficulties with raw data being interpreted as special characters). MIB entries that have variable length are sent padded to their maximum length. MCS shall not send an RPT command that results in a

response message whose length exceeds the maximum specified by this ICD. See Section 6 for an example of the use of this command.

Subsystems may also send unsolicited `RPT` response messages to MCS. An unsolicited `RPT` response has the same format as a normal `RPT` response, but the `REFERENCE` field is set to 999999999 (the maximum reference number). The `DATA` field contains the standard response fields followed by the MIB data for the reported entry or branch. This mechanism allows subsystems to proactively report status changes without waiting to be polled.

- "`SHT`" = Shutdown. The purpose of this command is to direct the system to shut down. If the `DATA` field is empty, then the shutdown should be "orderly"; e.g., tasks which are currently executing may be allowed to complete or be "gracefully" terminated. If the `DATA` field contains the string "`SCRAM`", then the subsystem should be shutdown as rapidly as possible; e.g., tasks which are currently executing should simply be abandoned.[4] If the `DATA` field contains the string "`RESTART`", then the subsystem should immediately restart after shutdown is complete. The data field "`SCRAM RESTART`" is permitted and has the expected effect.

The controlled subsystem shall respond to every message with a matching `DESTINATION` (or "`ALL`") with a "response message". This is demonstrated by example in Section 6. The response message shall be transmitted within 3 seconds of receipt of the associated message from MCS. If the `DESTINATION` field is not a match (or "`ALL`"), then the controlled subsystem shall ignore the message. The `DATA` field of a response message has the following structure:

1. `R-RESPONSE` [1 byte, ASCII]. This is the character "`A`" to indicate that the command was accepted, or the letter "`R`" to indicate that the command was rejected.
2. `R-SUMMARY` [7 bytes, ASCII]. This is MIB entry 1.1.
3. `R-COMMENT` [variable length, ASCII]. The definition of this field depends on `R-RESPONSE` and the message `TYPE`. If `R-RESPONSE` is "`R`", then this field shall be used to send error codes or log messages, as specified by the subsystem ICD or other subsystem design documents.

6 Command/Response Examples

For clarity in the following examples, single quotes (') are used in lieu of spaces and "`@`" is used to represent a byte of raw binary data.

6.1 PNG Command/Response

The following is an example of a `PNG` command sent from MCS to NDP. MCS sends a message packet containing the payload

```
NDPMCSPNG'''''1391'''0'54828'12345678'
```

which is interpreted as follows:

- `DESTINATION` is the NDP subsystem.
- `SENDER` is MCS.

[4] The intent of the "`SCRAM`" option is to provide a quicker method for shutting down the station to save time during integration, commissioning, and maintenance activities, when many power-up/power-down cycles may be required and there is no risk of data loss. (It is anticipated that the option will exist to simply cut power to a subsystem, but that this will be facilitated specifically through station PCD.)

- TYPE = "PNG"
- REFERENCE = 1391
- DATALEN = 0, so the DATA field is empty.
- MJD = 54828, so Dec 28, 2008 UT.
- MPM = 12345678. Dividing by 3600×1000 gives the number of hours past UT midnight; in this case, about 3.4.
- Mandatory space following the MPM field.
- The DATA field is empty.

In response, NDP sends the message

```
MCSNDPPNG’’’’’1391’’’8’54828’12345698’A’NORMAL
```

which is interpreted as follows:

- DESTINATION is the MCS subsystem.
- SENDER is NDP.
- TYPE = "PNG"
- REFERENCE = 1391 (same as the command message, so MCS can identify it)
- DATALEN = 8, so the DATA field is 8 bytes long.
- MJD = 54828 (same as the command message since the response occurs the same UT day)
- MPM = 12345698; this is NDP's estimate of when this response was sent.
- Mandatory space following the MPM field.
- The DATA field contains the 8-byte string "A’NORMAL", indicating that the associated command message was accepted, and that the value of MIB entry 1.1 (SUMMARY) is "NORMAL".

6.2 RPT Command/Response

The following is an example of an RPT command sent from MCS to NDP. This example assumes the MIB fragment shown in Table 2. MCS sends the message

```
NDPMCSRPT’’’’’1391’’’3’54828’12345678’B21
```

which is interpreted as a request for the data value associated with MIB index 2.1.

In response, NDP sends the message

```
MCSNDPRPT’’’’’1391’’’5’54828’12345698’A’NORMAL’’3.4
```

in which NDP is indicating that B21 = 3.4. Note that all 5 bytes of data value (per the specification of Table 2) are sent.

The following example is different only in that multiple values are requested simultaneously using a single branch index. MCS sends the message

```
NDPMCSRPT’’’’’1391’’3’54828’12345678’C22
```

In response, NDP sends the message

```
MCSNDPRPT’’’’’1391’’5’54828’12345698’A’NORMALPRR’7
```

in which NDP is indicating that D221 = “PRR” and E222 = 7. Again, note that all bytes of the data value (per the specification of Table 2) are sent. This is particularly important as MCS will simply count bytes to parse the `DATA` field into MIB entry data.

A References

[1] J. Craig, “Long Wavelength Array Station Architecture,” Ver. 2.0, Long Wavelength Array Memo 161, February 26, 2009.

[2] S. Ellingson, “MCS Subsystem Definition,” Ver. 2, Long Wavelength Array Engineering Memo MCS0004, Feb. 23, 2009.