E1.20 Review 3.1

Notes from the chair.

A number of comments received were in fact multiple comments. Where this occurred, the comment has been given a suffix in order to allow the group to respond thoroughly.

The public comments generated numerous new ‘intragroup comments’ as a result of discussion. These comments are numbered from 64 onwards and have been treated as if they were public comments.

1

Comment:

Why can’t have a PID SUPPORTED_VERSION, that returns the version of the E1.20 specification that the Responder supports with the GET command and the SET would tell the Responder what version of the E1.20 specification it supports. If the Responder never gets a SET Command it assumes E1.20-2015 and does not respond with Unicode or ACK_TIMER_HI_RES. If the Controller receives an NACK Unknown_Pid response type to the GET command it can assume the Responder supports only E1.20-2015 and ALL text will be ASCII. This method allows the specification to be modified in the future by updating the SET/GET commands to refer the latest released version of E1.20-XXXX. The Parameter Data would contain a single integer i.e. 2021 which indicates the specification version that is supported. It is defined that if a device supports the latest version it MUST support ALL previous versions.

Resolution: Withdrawn.

Notes: Group discussed at length and concluded the proposal was flawed. Commenter agreed and withdrew comment.

2

Comment:

Given NACK has a few meanings within the standard, can I suggest replacing "with a NACK with" to "with a NACK Response with". I see four instances in the current PDF. Or possibly the more verbose "with a NACK Reason Response with", but there seems to be slightly less consistency around that full term.

Resolution: Accept.

Notes: Actioned in document.

3

Comment:

Since the number of Sub-devices has been expanded to 65520 all of the other related specifications will need to be updated as they all list the maximum Sub-Devices as 512. E1.37-5, E1.37-7, E1.37-1 and E1.37-2. Additionally, All of the PIDs defined in E1.37-5, E1.37.7, E1.37-1 and E1.37-2 need to have their respective PID tables modified to define if "Packing Disallowed" as well if the PID is required as "Required Sub-device"

Note: Many PIDs require more than just an Index value with the GET command. i.e. E1.37-5, SLOT_DETAIL requires 2 parameter, "Slot # Requested" and "Number of Details". Therefore they cannot be used with the Packing PIDs.

Resolution: Reject.

Notes: All standards reference a specific version of E1.20 so this is not an issue.

4 Section 1.0

Comment:

Was there some sort of tracked changes/diff for this round of public review, or will that only start happening with new documents/future cycles (I can appreciate integrating it into stuff that's already in progress may be a challenge). I just wanted to check I haven't missed it somewhere.

Previous Comment 1

Resolution: Reject.

Notes: Commentor has not proposed a change. Diff copies of the last round were created and sent to commenter. Diff copies of this round have also been created.

5 Section 6.3.5

Comment:

"NOTE: A NACK Reason Code of NR_FORMAT_ERROR is allowed in discovery as defined in Table 6-7." versus "Responders shall not respond to valid DISC_MUTE and DISC_UN_MUTE requests with a NACK. Responders may respond to invalid DISCOVERY_COMMAND requests, excluding DISC_UNIQUE_BRANCH, with a NACK." is it only NR_FORMAT_ERROR, or any NACK?

Previous Comment 4 (6.3.5 Negative Acknowledge (RESPONSE_TYPE_NACK_REASON))

Resolution: Accept.

Notes: Edited the text below Table 6-7 to clarify.

6 Section 10.4.1

Comment:

So just to clarify on this comment, should e.g. the device model ID be different when a fixture is in bootloader mode or when it's been set back to factory defaults? If the dimmer module subdevice I query has just been reset at the point I cache that set of supported PIDs for all its friends, how am I supposed to know it supports many more PIDs once it's finished loading?

Previous Comment 16 (10.4.1 Get Supported Parameters (SUPPORTED_PARAMETERS))

Resolution: Reject.

Notes: Commentor has not proposed a change. The document makes it clear that in bootloader, a product has very limited functionality.

7 Section 10.4.2

Comment:

Just to pick back up on my array example, personally I'd consider a "variable length field" to be one that varies, like a string, whereas in the case of the "manufacturer that returns a PDL size of 24, with a UINT8, to represent 3 bytes of data.", it would be good to cover that too with a word like repeated/arrayed/multiple. I see Table A-15 mentions "Data is an array of unsigned bytes", but that's the only place in the entire standard the word array is used, also including it (or something equivalent) in the main 10.4.2 section would probably help.

Previous Comment 18 (10.4.2 Get Parameter Description (PARAMETER_DESCRIPTION))

Resolution: Reject.

Notes: The group accepts there are numerous issues with this PID. But that is the reason for E1.37-5.and the group did not feel there was value in trying to fix it here too.

8 Section 10.4.2

Comment:

The resolved text is "This field has no meaning for a Data Type of DS_BIT_FIELD or DS_STRING. <SNIP> This field has no meaning for non-integer Data Types.". It would probably be clearer if at least the two non-integer references were brought together sentence wise, or probably combined into one sentence.

Previous Comment 19 (10.4.2 Get Parameter Description (PARAMETER_DESCRIPTION))

Resolution: Accept.

Notes: Actioned in document.

9 Section 10.4.2

Comment:

Sorry, I think this one needs reopening, as Table A-15 now has UINT/INT64 which are not "non-integer Data Types.", but won't fit in the existing 32 bit field.

Previous Comment 20 (10.4.2 Get Parameter Description (PARAMETER_DESCRIPTION))

Resolution: Accept.

Notes: Clarification added to Data Type in 10.4.2

10 Section 10.4.7

Comment:

Fix the typo "For details of PACKED_PID_INDEX see Section 10.4.7. For details of PACKED_PID_SUB see Section 10.4.7." PACKED_PID_SUB is 10.4.6, but probably wants to be first.

Previous Comment 23 (10.4.7)

Resolution: Accept.

Notes: Actioned in document.

11 Section 10.4.7

Comment:

Add "Section 10.4.7: Get/Set Packed List of PIDs by Index (PACKED_PID_INDEX) - Type: Improvement First Item Requested / Encoded: Add a comment similar to PACKED_PID_SUB regarding alignment for e.g. DMX_PERSONALITY_DESCRIPTION “When the index field of the PID is less than 16-bits it shall be right justified.”

Previous Comment 38. (10.4.7 Get/Set Packed List of PIDs by Index (PACKED_PID_INDEX))

Resolution: Accept.

Notes: Actioned in document.

12 Section 10.4.8

Comment:

It would be good to add a note explicitly clarifying that requests may be non-contiguous, the TG seemed to agree when they reworked it, but the only mention is still the rather subtle "(For Example: A PID of NACK_DESCRIPTION means the range is the number of defined NACK Reason Codes). A value of 0xFFFF shall require the responder to return all available data."

Previous Comment 40 (10.4.8 Get/Set Packed List of PIDs by Index (PACKED_PID_INDEX))

Resolution: Accept.

Notes: Actioned in document.

13 Section 10.7.2

Comment:

Since the update linked to this comment, the standard no-longer gives any detail of what should happen to Lowest/Highest/Recorded during a set if they are supported, which seems a step backwards to me. It should at least clarify what gets reset during a reset and to what values (i.e. it's not just a reboot of the sensor). IMHO the previous text was fine, it just needed a clarification that it only applied if the respective field was supported.

Previous Comment 52 (10.7.2 Get/Set Sensor (SENSOR_VALUE))

Resolution: Accept.

Notes: Actioned in document.

14 Table A-13

Comment:

Both the comments and old comments don't appear to apply to this comment (which was about rate of data transfer), the old comments appear to be referring to comment 71. Given some of my more esoteric SI units got approved, it seems a bit odd this one got rejected, hence I wonder if it's an error. TLDR: Please add a bytes or bits per second, the equivalent

Table A-12 reference and extend the existing reference about 1000/1024 ""When a prefix is used with MEMORY, the multiplier refers to binary multiple. i.e. KILO means multiply by 1024.

Previous Comment 70 (Appendix A Table A-13: Sensor Unit Defines)

Resolution: Accept.

Notes: Actioned in document.

15 Table A13

Comment:

Some clarification and a far better example of my previous comment. Consider either a moving head with a couple of filters which need changing/cleaning every 1,000 or 10,000 hours of operation (and a sensor to indicate this run time, which may be independent of the overall fixture or lamp hours). Or a fog machine where it's desirable to implement a ""real world"" sensor value of estimate run time (like cars do now). Currently I have to use the unit of seconds for both, for the filters, I have to use 3,600 kilo seconds or 36 mega seconds respectively, which isn't very catchy. For the smoke machine, I might report up to 14400 deca seconds, but when it says 1080 deca seconds, do you know if it needs refilling before or after the show? For all the above scenarios, just being able to see 1,000/10,000/40 or 3 hours respectively is a lot easier to understand, and is both at a sufficient level of precision of the usage, and likewise probably an acceptable level of precision given the accuracy of the measurement.

Previous Comment 71 (Appendix A Table A-13: Sensor Unit Defines)

Resolution: Reject

Notes: The group feels that the protocol does the job of data transfer. This is an UI / implementation issue.

16 Table A16

Comment:

Another commented previously said ""DS_MAC" in Table A-16 should be renamed to "DS_ETHERNET_MAC" because there are different MACs for different transport media.". However this is confusing, as the same MAC address system is used for WiFi too, which is a different standard (hence not Ethernet). We should use the same terminology as in E1.37-2 which was INTERFACE_HARDWARE_ADDRESS_TYPE1 AKA the EUI-48 [EUI] hardware address of an interface. Or some improved compromise of that.

Previous Comment 82 (Table A-16)

Resolution: Accept in principle.

Notes: See document for changes.

17 Appendix C

Comment:

Sorry to be pedantic, but ""The Slot Index is a reference to the offset from the DMX512 Starting Address"", so in Appendix C, please replace the remainder of ""Indexes"", ""Manufacturer-specific IDs"" and ""Publicly defined IDs"" with ""Types"" or ""Slot Label IDs"" as well as dropping the Index from ""Slot Index Type"", so the names used here match the stuff in SLOT_INFO.

Previous Comment 94 (Appendix C Slot Info (Normative))

Resolution: Accept in principle.

Notes: See document for changes.

18

Comment:

RDM has now become very widely accepted. The current complaint is that Manufacturer Specific Pids often need a tool from that manufacturer to us. Perhaps the most important type is the ‘combo box’ pids where a range of numbers need a text enumeration to be meaningful. Examples of standard PIDs are PERSONALITY and PERSONALITY_DESCRIPTION.

In principle e1-37-5 PARAMETER_METADATA can solve this, but for light weight devices it will be cumbersome. A controller needs to download the entire meta data file in order to tell the user what decimal 234 means. If interleaving zero start code DMX, that would be very slow.

PERSONALITY and PERSONALITY_DESCRIPTION is a more elegant solution as the text for decimal 234 can be retrieved in one command.

I’m calling PERSONALITY_DESCRIPTION a ‘helper pid’ for PERSONALITY. I would like to include the concept of a helper pid in PARAMETER_DESCRIPTION. We have an unused field Type. This can be changed to an offset from the core pid for helper pid.

Example:

ManSpec PID = 0x8000

PERSONALITY_DESCRIPTION.Type = 0x7f.

Helper PID = 0x8000 + 0x7f = 0x807f.

This would make a huge difference to support for manufacturer specific pids.

Resolution: Accept in principle.

Notes: See document for changes. This also relates to E1.37-5 comment #2.

There was much discussion and concerns that we should not invent multiple ways of solving the same problem. The group reached two conclusions:

1) The E1.37.5 group should be encouraged to add a PID that allows a request for a filtered JSON response which reduces the processing load on small systems.

2) The core comment had value, particularly when related to low power hand-held devices and should be implemented (implemented in document).

19

Comment:

The JSON schema shown in the draft E137-5 document belongs HERE, in E1.20

NOT in E137-5.

Particularly as you are wishing to deprecate the use of PARAMETER_DESCRIPTION as described in this document. It is daft to update this document without any mention that the existing PARAMETER_DESCRIPTION PID is so broken, and currently not good for much more than returning the text description of the PID.

Notes: Previously discussed and rejected. The Parameter Description Metadata ONLY applies to manufacturer-specific PIDs. This metadata is not mandated or required for manufacturer-specific PIDs. This would also substantially slow down the process at these late stages as we currently have two separate groups each editing documents instead of bottlenecking into a single doc.

https://tsp.esta.org/tsp/meetingfiles/cpwg/BSR_E1-20_201x-6%20016.pdf

Resolution: Reject.

Notes: Previously discussed at length – group felt there was no need to change.

20a Section 6.2.7.1

Comment:

The term “Controller Port” is mentioned here (and in 3.1.2), but not clearly defined. (Not in appendix D).

A number of legacy controllers have incorrectly set this field to zero, and the revised standard should explicitly define what is expected of a responder in this case. Existing responders could have legitimately rejected such traffic, but market forces have required us to be lax, even though is has been the controllers at fault.

The term “command port” is already used in the standard to refer to physical ports. I believe the original intent of this field was to embed in the data stream an indication of which physical port the data originated from. There are occasions where this is admirable and useful. Equally there are many cases (for example a simple splitter that does not have a processor generating new data for each physical output) where this is not possible.

The standard should reflect this, and state that controllers shall set the field in the range 1-255, but that responders shall interpret 0 as a valid value and treat it as if it were 1.

Resolution: Accept in part.

Notes:

Changes:

- Change text to Controller’s Command Port.
- Command Port capitalise in all document
- Control/ler Port capitalise in all document

20b Section 6.2.7.1

See comment 21

21 Section 6.2.7.1

Comment:

I thought they were going to allow the PORT ID field to be 0-255, still listed as 1-255?

Resolution: Reject.

Notes: This would make compliant devices not compliant.

22 Section 6.2.8.1

Comment:

What happens if a Responder receives a Controller Flag that contains a non-supported bit as defined in table 6-3a? Should the message be rejected (Data Range Error/Format Error) or ignored with no response or ignored with a normal response?

If the payload of a controller generated message contains the Unicode bit and the Responder does not support Unicode, should the responder respond with a NACK reason or ignore the bit and respond normally? What NACK reason?

Section 6.2.8.1: If the payload of a controller generated message contains the Unicode bit and the response does not contain a text field, should the responder reject the message with a NACK reason or ignore the bit in the Controller Flag. If rejected what is the NACK reason?

Resolution: Accept.

Notes: Actioned in document. Any undefined controller flags set – NR_FORMAT_ERROR.

23 Section 6.2.8.1

Comment:

Controller Flags

This section needs to define what a responder should do on receipt of non-supported bits being set.

This section needs to define what a responder should do on the Unicode bits being set and the responder not supporting Unicode.

This section needs to define what a responder should do on the Unicode bits being set and the responder response not containing a text field.

Reject with NACK – if so which one, ignore with no response ? I suggest there is an NACK: UNSUPPORTED_CONTROLLER_FLAG reason added.

However I would also recommend a serious appraisal of the suggestions by others to the use of a PID SUPPORTED_PROTOCOL_VERSION. A GET would allow a controller to ascertain what the latest version of the standard a Responder is capable of, and a SET would inform the responder as to the controller capability.

Resolution: Accept.

Notes: Actioned in document.

24 Section 6.3.5

Comment:

If a responder supports ACK_TIMER_HI_RES why does it need to tell the Controller and therefore why does the Controller need to give the Responder permission to use it? If the Controller receives a response type of ACK_TIMER_HI_RES then the Controller should properly compute the delay time just as it did for the normal ACK_TIMER. Passing the HI_RES bit back and forth in the Control Flag seems redundant.

Resolution: Reject and commenter withdraws.

25 Section 6.3.5

Comment:

(RESPONSE_TYPE_NACK_REASON)"

Change: "The format of the MDB of the response message is defined as follows:"

To: "The format of the MDB of the response message is shown in the following example:"

This way, the text doesn't imply fixing the CC to GET_COMMAND_RESPONSE.

Resolution: Accept.

Notes:

Change table of response example to show CC = GET command response / set command response.

26 Section 6.3.4

Comment:

(RESPONSE_TYPE_NACK_REASON)"

The RESPONSE_TYPE_NACK_REASON shows the response as using GET_COMMAND_RESPONSE. This could be just an example, or it could mean to always use GET_COMMAND_RESPONSE, even for SET_COMMAND requests. If it's always supposed to be GET_COMMAND_RESPONSE, then I think we should add a sentence: "A GET_COMMAND_RESPONSE shall always be used as the Command Class." If this is not true, I would change the "(CC)" table element to "GET_COMMAND_RESPONSE or SET_COMMAND_RESPONSE".

Resolution: Accept in principle.

Notes: Section re-drafted.

27 Section 7.5

Comment:

Typo – should read Discovery Unique Branch

Resolution: Accept.

Notes: Actioned in document.

28 Section 9

Comment:

The example in this section refers to dimmer racks, the use of which is becoming increasingly rare with the advent of LED technology and the demise of incandescent fixtures.

The use case for increasing the number of sub-devices is not given in this revision.

The example should be revised or added to in order to make the document relevant to the likely use case.

I am firmly of the view that allowing 65320 sub-devices is the wrong decision. Since there is no way of restricting this to RDMNet applications, the existence of so many sub-devices at the RS485 level could completely swamp the available bandwidth.

A legacy controller should be capable of operating with a responder compliant with this new standard, albeit without the benefit of any new features. Thus it should be possible to discover and set the address of any fixture. A controller written to the existing standard could legitimately reject a DEVICE_INFO declaring more than 512 sub-devices, and thus not be able to control or configure the basics of a target device.

Being able to handle this many sub-devices puts an unnecessary burden on controllers.

Resolution: Reject.

Notes: This number has been increased to support more modern equipment such as video walls.

We made the sub-device range consecutive. This makes the situation better for bandwidth.

- 512 and non-consecutive was old system
- 16-bit and consecutive is the new system and a big improvement for pixel mapping.

29 Section 9.2.1

Comment:

Why is there a need for 65320 Sub-Devices?

Resolution: Reject.

Notes: This number has been increased to support more modern equipment such as video walls.

Group decided to make sub-device range consecutive. This makes the situation better for bandwidth.

Clarification:

- Previously Sub-devices were up to 512 and non-consecutive.
- Now, Sub-devices are16-bit and consecutive.

30 Section 9.2.2

Comment:

The first and second paragraphs are confusing, as they discuss both the number of sub-devices and the means of getting the list of supported parameters. I suggest moving para1 sentence 2 to become a third paragraph – i.e. after the discussion about how many there are and how they find need to be scanned.

Resolution: Accept in part.

Notes: Second sentence has been deleted.

31 Section 9.2.3

Comment:

This section does not mention a number of required Messages (such as DEVICE_INFO) mentioned in table A-3. The section should at least reference Table A-3, as this table has been enhanced since the earlier revision of the standard

Resolution: Accept.

Notes: Actioned in document.

32 Section 10.1.1

Comment:

In the case of a Unicode response the Responder could use the Response Type of the ACK_UNICODE to indicate the text field is Unicode, otherwise the normal ACK indicates the text field would be ASCII. If the Controller does not support E1.20-20xx then the responder MUST respond with ASCII text and an ACK conforming to the E1.20-2015 specification.

By not changing the meaning of the Message Count, field there should be minimum changes to Controller/Responder code. Since we only need to add ACK_UNICODE as a new Response Type (we already added ACK_TIMER_HI_RES) we need minimal changes in the Controller and Responder code. ACK_UNICODE could be used equivalent to ACK even if no text fields are present OR the ACK_UNICODE could be restricted to only responses that contain text fields. This method keeps the existing structure of the E1.20 specification intact by NOT adding special meaning bits within the message structure but by using PIDs to interface with the Controllers and Responders. Keeping in mind that RdmNet relies on the E1.20 specification changing the Message structure could cause adverse issues and by using PIDs and Response Types to add functionality is more keeping with the intent of the original E1.20 specification and subsequent E1.37-X add on specifications

Resolution: Reject.

Notes: Group felt the flags implementation was a god solution offering future options.

33 Section 10.3.1

Comment:

This section uses the term “subscribed sensor”, but there is no mention elsewhere in the document as to what this is.

The paragraph under the PID example that reads “The status type requested is …. Or as allowed in table 10-1” is still confusing, as Table 10-1 is a table of required responses. The correct reference should be Table A-4.

Resolution: Accept.

Notes: Actioned in document.

34 Section 10.3.22

Comment:

The note about the earlier standard not allowing use of STATUS_NONE should really be in 10.3.1 as it relates to the outgoing GET.

Indeed the use of STATUS_NONE is to be preferred if you just want to poll for Queued Messages and not be swamped by possible error, advisory or warning messages.

Resolution: Accept in part.

Notes: Second sentence deleted

35/36 Section 10.3.6

Comment:

QUEUED_MESSAGE_SENSOR_SUBSCRIBE

Incorrectly referenced in table A-3?

The section 10.3.6 defines the pid as QUEUED_MESSAGE_SENSOR_SUBSCRIBE, however in table A-3

it is defined as QUEUED_MESSAGE_PID_SUBSCRIBE. I assume it should be as defined in 10.3.6

Resolution: Accept.

Notes: Actioned in document.

37 Section 10.4.3

Comment:

Get supported Parameters Enhanced

PID Support - incorrect forward reference for PACKED_PID_SUB, which is 10.4.6 not 10.4.7.

Consider referencing these in the order in which they appear later in the document. (ie SUB, then INDEX)

Resolution: Accept.

Notes: Actioned in document.

38 Section 10.4.3

Comment:

There is an issue with table numbering – why is there a table 10-1a?

Resolution: Reject.

Notes: It was felt that renumbering tables would cause confusion.

39 Section 10.4.3

Comment:

Is the "Non-identical Sub-Device" flag actually a per-subdevice rather than per PID flag? i.e. if it's set to 1 on any PID, should it be set to 1 for all of them? If so, why is it repeated multiple times in the data, not once as a non-repeating set of flags at the start of the data? The comment below Table 10-1a seems to imply it would be set to true for DMX_PERSONALITY for example if each subdevice had a different set of personalities available (for example, which doesn't seem to be disallowed by the standard). Whereas the descriptions above the table "determine if all sub-devices have the same list of supported parameters" and "If the "Non-identical Sub-Device" flag is not set for any parameter in a Supported Parameters Enhanced response, then each Sub-Device shall have an identical response for SUPPORTED_PARAMETERS_ENHANCED to each other Sub-Device." all appear, by my reading at least, to be related to the PID list in SUPPORTED_PARAMETERS_ENHANCED rather than what you get when you query the listed PID.

Resolution: Accept in principle.

Notes. This section was reworked.

40/41 Section 10.4.5

Comment:

Get Nack Reason Description

This is missing a corresponding PID to get a list of valid manufacturer specific NACK reason codes.

I don’t want to have to poll all possible values to know which ones are valid and which ones are erroneous.

I have no idea how to know what Custom NACK Reason to ask for by number using the new PID

NACK_DESCRIPTION. I can't find anywhere the device tells me what the custom numbers are. Do I what until

I receive a NACK with a reason greater than 0x8000 and then ask for the description? How do I know it is not

a mistake reason?

It seems there should be a PID similar to SUPPORTED_PARAMETERS, called MANUFACTURER_NACK_REASONS that returns a list of manufacturer defined NACK Reasons. Then just like the PID PARAMETER_DRECRIPTION the Responder can be queried for the descriptions using NACK_DESCRIPTION from the list returned

Resolution: Accept in principle.

Notes: Actioned in document.

42 Section 10.4.6

Comment:

Get Packed List for Sub-Devices

I found the terminology in this section confusing, as to when “sub-device means sub-device not including root, and all devices, ie including root.” Elsewhere we refer to the Root device, and Sub-Devices. All of a sudden in this section, you are using the term Sub-Device to include the root device.

Consider changing the field names referring to First Sub-devices, Number of Sub-devices, to

First Device, and Number of Devices.

The text for First Device Requested would then read

“ The device encoded for entry x of the response packet. A value of 0x0000 indicates the root device. Values in the range 0x0001-0xFFF0 indicate a sub-device.

It would then be clearer why the Number of Devices is in the range “0x0001 to 0xFFF1 (one more than the number of sub-devices described elsewhere in the document as 0xFFF0)

Resolution: Accept.

Notes: Actioned in document.

43 Section 10.4.6

Comment:

Get Packed List for Sub-Devices

The example GET:PID shows the number of sub-devices requested as 0xFFFF, but the earlier section (top of page 71) states that this field shall be in the range 0x0001-0xFFF1. By my reckoning the example should have 0x0003 in this field.

Resolution: Accept in part.

Notes: Commentor is correct, but problem was fixed by allowing All_Call in quantity field.

44 Section 10.4.6

Comment:

Can a responder implement this PID while still behaving as per the previous version of this standard with non-contiguous sub device numbers? If no, please explicitly clarify that. If it can, say I have sub devices at 0x0001, 0x0003 and 0x0005, and I do a PACKED_PID_SUB for the first as 0x0001 and request three sub devices, does the whole thing NACK due to "if any of the packed replies would generate a NACK response, the responder shall return a NACK Response" which would be triggered by my sub device out of range, or does it return the intervening devices with a PDL of 0, or does it only return the ones that exist? A clarifying note would be helpful in all regards. If it's only supposed to handle contiguous ones, the encoding of all item values seems a bit redundant. If non-contiguous is allowed, having an example showing this too would be very beneficial.

Resolution: Reject.

Notes: The pid would not be used with older versions of the standard.

45 Section 10.4.6

Comment:

If you consider say GET SELF_TEST_DESCRIPTION, which doesn't explicitly require contiguous self test numbers, so say I have self tests at 0x01, 0x03 and 0x05, and I do a PACKED_PID_INDEX for the first as 0x01 and request three items, does the whole thing NACK due to "if any of the packed replies would generate a NACK response, the responder shall return a NACK Response" which would be triggered by my data out of range, or does it return the intervening numbers with a PDL of 0, or does it only return the ones that exist? A clarifying note would be helpful in all regards. If the latter, I don't know if there is some way the PD table can better represent this (the sub device one is clearer, if it had item for entry x/x+1 etc). If it's only supposed to handle contiguous ones, the encoding of all item values seems a bit redundant.

Resolution: Accept in part.

Notes: If responder cannot usefully respond to a packed request, it should NACK reason. NR_ERROR_IN_PACKED_LIST_TRANSACTION. Clean up edits implemented in document.

46 Section 10.6.4

Comment:

When we wanted to implement them we noticed that several controller manufacturers did not understand what the standard requires. It is not easy to understand but easy to interpret. You have to go throug a long video on Ethernet about this subject to get some clarifications. It is all due to a lack of a clear list of examples that we would like to propose. Those examples are from a real implementation and work with many well known controllers

The following example is from our latest product SULLY. Slot Info is currently implemented and works with several lighting consoles from leading manufacturers.

We wanted to show the issue as RDM Integrity shows it as a lot of manufacturers are using this test tool. Unfortunately RDM Integrity is wrong when you have a 16 bit slot and it creates a lot of issues. Even for us it is difficult to explain clearly what to do. The standard in this occasion uses words and table that are misleading and many controllers fell into the trap. The issue is often in 16 bit. Our example:

Mode 6: Followspot16b

Dimmer

Dimmer fine

Master

Master fine

Strobe duration

Strobe speed

Response time

Control mode

Slot Offset [0x0000, (0)]

Slot Type [0x00, (0)]

Primary Slot Slot Label ID [0x0001, (1)], Intensity

Slot Offset [0x0001, (1)]

Slot Type [0x01, (1)] Secondary Fine

RDM Integrity is wrong by showing this : Slot Label ID [0x0000, (0)], Slot Label ID Undefined Entry [0x0000, (0)] It is a secondary slot, so the Slot Label ID should be a reference to the offset of the slot that is its primary.

Slot Offset [0x0002, (2)]

Slot Type [0x00, (0)] Primary Slot Slot Label ID [0x0002, (2)], Intensity Master

---

Slot Offset [0x0003, (3)]

Slot Type [0x01, (1)] Secondary Fine

RDM integrity is wrong: Slot Label ID [0x0002, (2)], Intensity Master

---

Slot Offset [0x0004, (4)]

Slot Type [0x00, (0)] Primary Slot Slot

Label ID [0x0404, (1028)], Strobe

Slot Offset [0x0005, (5)]

Slot Type [0x00, (0)] Primary Slot

Slot Label ID [0x0404, (1028)], Strobe

Slot Offset [0x0006, (6)]

Slot Type [0x00, (0)] Primary Slot

Slot Label ID [0x0503, (1283)], Fixture Speed

Slot Offset [0x0007, (7)]

Slot Type [0x00, (0)] Primary Slot

Resolution: Accept:

Notes:

- Table C1 change ST_SEC_INDEX to ST_SEC_QUANTUM
- Table C1 change ST_SEC_INDEX_ROTATE to ST_SEC_QUANTUM_ROTATE

47 Section 10.6.5

Comment:

When we wanted to implement them we noticed that several controller manufacturers did not understand what the standard requires. It is not easy to understand but easy to interpret. You have to go through a long video on Ethernet about this subject to get some clarifications. It is all due to a lack of a clear list of examples that we would like to propose. Those examples are from a real implementation and work with many well known controllers

The following example is from our latest product SULLY. Slot Info is currently implemented and works with several lighting consoles from leading manufacturers.

We wanted to show the issue as RDM Integrity shows it as a lot of manufacturers are using this test tool. Unfortunately RDM Integrity is wrong when you have a 16 bit slot and it creates a lot of issues. Even for us it is difficult to explain clearly what to do. The standard in this occasion uses words and table that are misleading and many controllers fell into the trap. The issue is often in 16 bit. Our example:

Mode 6: Followspot16b

Dimmer

Dimmer fine

Master

Master fine

Strobe duration

Strobe speed

Response time

Control mode

Slot Offset [0x0000, (0)]

Slot Type [0x00, (0)]

Primary Slot Slot Label ID [0x0001, (1)], Intensity

Slot Offset [0x0001, (1)]

Slot Type [0x01, (1)] Secondary Fine

RDM Integrity is wrong by showing this : Slot Label ID [0x0000, (0)], Slot Label ID Undefined Entry [0x0000, (0)] It is a secondary slot, so the Slot Label ID should be a reference to the offset of the slot that is its primary.

Slot Offset [0x0002, (2)]

Slot Type [0x00, (0)] Primary Slot Slot Label ID [0x0002, (2)], Intensity Master

---

Slot Offset [0x0003, (3)]

Slot Type [0x01, (1)] Secondary Fine

RDM integrity is wrong: Slot Label ID [0x0002, (2)], Intensity Master

---

Slot Offset [0x0004, (4)]

Slot Type [0x00, (0)] Primary Slot Slot

Label ID [0x0404, (1028)], Strobe

Slot Offset [0x0005, (5)]

Slot Type [0x00, (0)] Primary Slot

Slot Label ID [0x0404, (1028)], Strobe

Slot Offset [0x0006, (6)]

Slot Type [0x00, (0)] Primary Slot

Slot Label ID [0x0503, (1283)], Fixture Speed

Slot Offset [0x0007, (7)]

Slot Type [0x00, (0)] Primary Slot

Slot Label ID [0x0502, (1282)], Fixture Control

Resolution: Accept in part.

Notes: Section re-drafted.

48 Section 10.11.4

Comment:

Get/Set Perform Self Test (PERFORM_SELFTEST)SELFTEST/

The standard says that you may return pass/fail status or other information by queuing a Status Message response. We implemented this but the reality is that very few controllers display Status_Message especially the major brands (lighting consoles for example).

- We created a Manufacturer Pid to display the result. The issue with this solution is that the status has to be seen in other screen or page and not right next to Selftest PID.

- Would it be possible to create an official PID (Value: 0x1022) to display the pass/fail status? We found that the controller has to check wether the 254 potential tests are present or not and it is relatively long.

- Would it be possible to declare the number of tests available in the description or returning something to tell the controller that there is no more selftests (end of enumeration)?

Resolution: Reject

Notes: Commentor highlights that numerous controllers implement status message badly or not at all. Group feels the controllers must improve – but this is not a reason to design a new way of responding.

Whilst this comment was rejected, it did generate a number of document changes and clarifications.

49 Section 11.1

Comment:

Polling Intervals

Para 2 – the Note should be on a new line.

What is the justification for restricting polling to intervals of 5 seconds?

Either justify the time interval or remove the advice.

Resolution: Accept.

Notes: 5s para deleted. Implemented in document.

50 Table A-3

Comment:

The PID value for SUPPORTED_PARAMETERS_ENHANCED is incorrect

Resolution: Accept.

Notes: Actioned in document.

51 Table A-3

Comment:

The section 10.3.6 defines the pid as QUEUED_MESSAGE_SENSOR_SUBSCRIBE, however in table A-3 it is defined as QUEUED_MESSAGE_PID_SUBSCRIBE. I assume it should be as defined in 10.3.6

Resolution: Accept.

Notes: Actioned in document.

52 TableA-3

Comment:

Page 120 Support Parameters Enhanced has the same PID # as Supported Parameters

Resolution: Accept.

Notes: duplicate 50.

53 Table A-3

Comment:

Lamp ON Mode formatting issue

Resolution: Accept in principle.

Notes: Actioned in document.

54 Table A-3

Comment:

PIDS 0xFFE0-0xFFFF – please drop the reference to “Future”. These are already in use for development purposes.

Resolution: Accept.

Notes: Actioned in document.

55 Table A-4

Comment:

"STATUS_GET_LAST_ MESSAGE" has an extra space after the last underscore. In general, running a search for "_<space>" would be useful before publication (although it may throw up the odd red herring in tables where things wrap), as I don't think this is the first one I've reported.

Resolution: Accept.

Notes: Actioned in document.

56/57/58 Table A-12

Comment:

Missing text/description for entry 0x23??

Resolution: Accept.

Notes: Actioned in document. Deleted and renumbered above.

59 Table A-14

Comment:

Negative SI multipliers for bits and other integer units, it would be good to have a note flagging these as discouraged or disallowed, my half joking suggestion of bits is the only way you could have less than a byte. Probably likewise for UNITS_NONE, if it was some sort of counter it shouldn't have a fraction. See related https://www.rdmprotocol.org/forums/showthread.php?t=1326

Resolution: Reject

Notes: It doesn't break it.

60a Table A-14

Comment:

The note about binary multiples and memory, should it be changed to SENS_MEMORY as per A-12. Or perhaps better yet, rewrite all those notes to be along the lines of "When a prefix is used with UNITS_BYTE, the multiplier..." and similar for the other notes. See related https://www.rdmprotocol.org/forums/showthread.php?t=1326

Resolution: Reject

Notes: The group disagreed as this would confuse units and sensors.

60b Table A-14

Comment:

The note about binary multiples and memory, should it be changed to SENS_MEMORY as per A-12. **Or perhaps better yet, rewrite all those notes to be along the lines of "When a prefix is used with UNITS_BYTE, the multiplier..." and similar for the other notes. See related https://www.rdmprotocol.org/forums/showthread.php?t=1326**

Resolution: Accept.

Notes: This comment also led to discussion of decibels. The group then added the following to A12.

- SENS_LOG_RATIO
- SENS_LOG_RATIO_VOLTS
- SENS_LOG_RATIO_WATTS
- SENS_LOG_RATIO_METER

61 Table A-15

Comment:

Given some recent comments, is it worth adding nominative references to the table for URL/MAC/IPv4/IPv6 (as already present in other docs) to avoid any confusion about which MAC is being referred to?

Resolution: Accept.

Notes: Actioned in document. Edited table A15 for clarity and added normative reference.

62 Appendix D

Comment:

We need to tighten our definition of relationship between prefixes and digital data types. Eg. Kilo = 1000 vs 1024.

Resolution: Accept.

Notes: Actioned in document. Added 2^{10} to document

63 Appendix D

Comment:

Add a definition for a port. Port: A means of physical connection of an RDM signal to a device.

Resolution: Accept.

Notes: Actioned in document.

64 Appendix D

Comment:

Section "6.3.3 Acknowledge Timer (RESPONSE_TYPE_ACK_TIMER)": "Any intervening messages other than STATUS_MESSAGE would show an incremented message count." should pluralize "STATUS_MESSAGES" because that's the name of the PID.

Resolution: Accept.

Notes: Actioned in document.

65 Section 6.3.3

Comment:

Section "6.3.3 Acknowledge Timer (RESPONSE_TYPE_ACK_TIMER)": Since a responder is only required to add data to its message queue when an ACK_TIMER message is ready to be retrieved, are there any sort of guarantees that a follow-up QUEUED_MESSAGE request will return the response first, in the presence of other potential queued messages? For example, there could be a message already in the queue because of some physical user

input to the device, changing some state and requiring some status update. I think it would be useful to add a sentence or two to this section clarifying that the controller should keep sending a GET QUEUED_MESSAGE until it sees the relevant response to an ACK_TIMER. I'd put this after the example and say something like, "Note that the future response associated with the ACK_TIMER message may not be the first message in the queue. A controller may wish to keep requesting queued messages until the desired response is encountered." This is assuming that it's possible for GET QUEUED_MESSAGE to retrieve messages that come before the ACK_TIMER-affiliated response. Thoughts?

Resolution: Accept.

Notes: Actioned in document.

66 Table A3

Comment:

"Table A-3: RDM Categories/Parameter ID Defines" (Status Collection section): There are several rows where "Comment" text could be changed to one line. Specifically, the "See Table A-4" comments.

Resolution: Accept.

Notes: Actioned in document.

67 Table A3

Comment:

"Table A-3: RDM Categories/Parameter ID Defines": In the LAMP_ON_MODE row, "Table A-9" should have the same size font and could stand to be on one line too.

Resolution: Accept.

Notes: Actioned in document.

68 Section 6.3.4

Comment:

Section "6.3.4 Negative Acknowledge (RESPONSE_TYPE_NACK_REASON)": Change: "The format of the MDB of the response message is defined as follows:" To: "The format of the MDB of the response message is shown in the following example:" This way, the text doesn't imply fixing the CC to GET_COMMAND_RESPONSE.

Resolution: Accept.

Notes: Actioned in document.

69

Comment:

Sub-Device fields in SUPPORTED PARAMETERS and SUPPORTED_PARAMETERS_ENHANCED are different than others that allow root+subdevices. The text should instead be, "0x0000 (Root) or 0x0001-0xFFF0".

Resolution: Accept.

Notes: Actioned in document.

70

Comment:

PARAMETER_DESCRIPTION and NACK_DESCRIPTION only allow Root for the subdevice, but Table A-3 says that these PIDs are required for subdevices (when there's manufacturer-specific PIDs). Change: Sub-Device field in these parameter descriptions to include all subdevices.

Resolution: Accept.

Notes: Actioned in document.

71

Comment:

PARAMETER_DESCRIPTION: Suggestion: Use "PD Size" or "PDL" instead of "PDL Size"?

Resolution: Accept.

Notes: Actioned in document.

72

Comment:

PARAMETER_DESCRIPTION: Add periods after "Min" and "Max" in "Min Valid Value" and "Max Valid Value".

Resolution: Accept.

Notes: Actioned in document.

73

Comment:

DMX_PERSONALITY_DESCRIPTION: Change "(3 + Number of Bytes sent)" to "(3 + Description size)". If it's decided not to change, lower-case the 'B' to be consistent.

Resolution: Accept.

Notes: Actioned in document.

74

Comment:

Similar for SLOT_DESCRIPTION: Change "(2 + Number of bytes being sent)" to "(2 + Description size)".

Resolution: Accept.

Notes: Actioned in document.

75

Comment:

SLOT_INFO: Change all the "Slot index 0", "Slot index n", etc. to "Slot offset 0". There's two indexes here: an index into the slots array and the index into the packet. Using "offset" matches more closely with the text. If we decide not to change the text, at least change "n" to "N" to be consistent with other lists.

Resolution: Accept.

Notes: Actioned in document.

76

Comment:

DEFAULT_SLOT_VALUE: Either: Change "Slot index 0" et al to "Slot offset 0". Or: Remove the "Slot index 0" et al lines and change "0", "N", etc. to "Slot offset 0", etc.

Resolution: Accept.

Notes: Actioned in document.

77

Comment:

REAL_TIME_CLOCK should have seconds range from 0-60 to account for leap seconds. I would argue that Hours=24 is more esoteric than leap seconds where Seconds=60, so if Hours=24 is there, the leap seconds should be too.

Resolution: Accept.

Notes: Actioned in document.

78

Comment:

Clarify how IDENTIFY_DEVICE should behave across both warm and cold resets, per common use. Specifically, recorded sensor values, the IDENTIFY value, other state, etc. It may be the case that we don't want to specify anything more than what's already there: the Mute state.

Resolution: Reject.

Notes: Group concluded this should be left to the manufacturer.

79

Comment:

Additional fix related to IDENTIFY_DEVICE (Section 10.11.1):

Sentence error (last word): "The current state of a responders identification status may be obtained using a GET:smessage."
* "responders" should be "responder's"
* Fix "GET:smessage" to "GET:IDENTIFY_DEVICE"

SET SENSOR_VALUE: Clarify whether this just resets the sensor itself or clears all the min/max/recorded values, per common use. If rejected, please, in the explanation, describe the correct or common behaviour.

Resolution: Accept.

Notes: Actioned in document.

80

Comment:

SET SENSOR_VALUE when the sensor number is 0xFF: clarify what the SET_RESPONSE values should be. (See: https://www.rdmprotocol.org/forums/showthread.php?t=182) If rejected, please, in the explanation, describe the correct or common behaviour.

Resolution: Accept.

Notes: Actioned in document.

81

Comment:

In title, "7.5 Discovery Unique Branch Message (DISC_NIQUE_BRANCH)", change to "DISC_UNIQUE_BRANCH".

Resolution: Accept.

Notes: Actioned in document.

82

Comment:

QUEUED_MESSAGE_SENSOR_SUBSCRIBE: There are three places in this section where "QUEUED_MESSAGES" should be "QUEUED_MESSAGE", per the name of the PID elsewhere.

Resolution: Accept.

Notes: Actioned in document.

83

Comment:

Sub-Devices section: SET commands (eg. DMX_PERSONALITY) sent to SUB_DEVICE_ALL_CALL: What if some subdevices can set the variable but some cannot (for example, unsupported personality)? Should just those devices that support the change get changed and the others not, with no error being returned/some error being returned? Or, should no subdevices get changed and an error returned? What is common practice? The spec isn't clear on this. I suggest adding a clarification sentence or two to this section, I just don't know what to suggest because I'm not clear on the behaviour. Note: it may be the case that "controllers won't do this today", but that's not an appropriate way to write responder software; it needs to account for all cases, even bad ones, facing a badly-written controller. Another option is to change this to be "externally configurable" between: 1. Set some and return error, 2. Set some and return no error, or 3. Set none and return error. Nos. 1 and 2 might be appropriate for nearly homogenous subdevices. What is the most common behaviour for controllers today?

Resolution: Accept in part.

Notes: Group felt that is any sub-device could not accept the Set, done should. New para 4 added.

84

Comment:

In SUPPORTED_PARAMETERS, it states, "Note: The list of Supported Parameters may change when a device transitions from boot-loader to normal operation, after a personality change, or a SET: FACTORY_DEFAULTS." This means that a personality change can change device behaviour. Wouldn't this imply that a "Personality" is no longer just a "DMX" or "DMX512" concept? I suggest changing the name "DMX512 Personality" to just "Personality" where mentioned, for example in Device Info, or the "DMX_PERSONALITY" and "DMX_PERSONALITY_DESCRIPTION". They would be renamed "PERSONALITY" and "PERSONALITY_DESCRIPTION". This wouldn't affect existing code because the PID doesn't change, only the names in the text.

Resolution: Reject.

Notes: Group felt that for historical reasons, this will just confuse.

85

Comment:

SUPPORTED_PARAMETERS_ENHANCED: Font style in GET_COMMAND_RESPONSE table for "SUPPORTED_PARAMETERS_ENHANCED" needs to be consistent.

Resolution: Accept.

Notes: Actioned in document.

86

Comment:

SUPPORTED_PARAMETERS_ENHANCED: The "Non identical sub-device data is supported" flag should only have meaning for a Root device.

Resolution: Accept.

Notes: New text: “The controller shall ignore the "Non-identical Sub-Device" flag if receiving it from anything other than the Root Device. Sub-devices shall set the "Non-identical Sub-Device" flag to zero.

87

Comment:

DMX_START_ADDRESS: Clarify what happens when a start address is SET for a device having a footprint of zero. There are two options: don't return an error and just set the internal state of the responder, and the second option is to send an error. The problem with returning NR_UNKNOWN_PID is that the PID might be supported for other personalities having a non-zero footprint and that might confuse the controller, so might a better option be NR_DATA_OUT_OF_RANGE, if an error were to be sent? If rejected, please, in the explanation, describe the correct or common behaviour.

Resolution: Reject.

Notes: The document does address this scenario. 0xffff is returned. Group feels this is an unambiguous response.

88

Comment:

DS_INT64 type in Table A-15: Change to "0x0A" from "0x0a".

Resolution: Accept.

Notes: Actioned in document.

89

Comment:

DS_GROUP in Table A-15: Remove final period from description, for consistency.

Resolution: Accept.

Notes: Actioned in document.

90

Comment:

DS_BIT_FIELD in Table A-15: Hyphenate "bit-packed".

Resolution: Accept.

Notes: Actioned in document.

91

Comment:

Table A-13: Change everything to plural, for consistency: AMPERES, OHMS, WATTS, KILOGRAMS, NEWTONS, JOULES, PASCALS, SECONDS, DEGREES, STERADIANS, CANDELAS, LUMENS, BYTES, etc. This would match how they're described and how the sensor units are named, for example, "The units are in amps." Keep "IRE" singular, though, because that's how it's used, for example "-43 IRE".

Resolution: Accept.

Notes: Changed to singular for consistency

92

Comment:

Table A-13: Change singular DECIBEL units to plural, for consistency: UNITS_DECIBELS, UNITS_DECIBELS_VOLTS, UNITS_DECIBELS_WATTS.

Resolution: Accept.

Notes: Changed to singular for consistency

93

Comment:

Table A-13: Remove SENS_CONTACTS from UNITS_NONE row OR say that it's an example, because "CONTACTS" isn't the only thing that might want UNITS_NONE.

Resolution: Reject.

Reason: The column title says example – implicitly this is not exhaustive.

94

Comment:

Table A-13: UNITS_METERS: Change SENS_LENGTH_POSITION to SENS_LENGTH.

Resolution: Accept.

Notes: SENS_LENGTH_POSITION was a typo, fixed.

95

Comment:

Table A-13: References in SENS table (Table A-12) are missing: SENS_CHROMINANCE, SENS_LOG_RATIO,
SENS_LOG_RATIO_VOLTS, SENS_LOG_RATIO_WATTS, SENS_LOG_RATIO_METER, and SENS_PERCENT.

Resolution: Reject.

Notes: Reference is purely informative.

96

Comment:

Table A-13: Add plain VOLTS and AMPERES units. There's no WATTS_DC, WATTS_AC, etc. Or, change UNITS_VOLTS_DC and UNITS_AMPERES_DC to UNITS_VOLTS and

UNITS_AMPERES. What if you're doing instantaneous measurements and you don't know if it's DC or AC? Specifying AC or DC isn't really a job for the units.

Resolution: Reject.

Notes: AC / DC are not units.

97

Comment:

Table A-13: Change "AMPERES" to "AMPS".

Resolution: Reject.

Notes: This is the correct SI unit.

98

Comment:

Table A-13: Change "CENTIGRADE" to "CELSIUS". This appears to be the accepted international unit. According to https://en.wikipedia.org/wiki/Celsius#Centigrade_versus_Celsius, "the 9th meeting of the General Conference on Weights and Measures and the Comité International des Poids et Mesures (CIPM) formally adopted 'degree Celsius' in 1948 for the degree of temperature." See also: https://www.bipm.org/en/committees/cipm/cipm-1948.html

Resolution: Accept.

Notes: Actioned in document.

99

Comment:

Table A-6: Capitalize first letters of all the comments.

Resolution: Accept.

Notes: Actioned in document.

100

Comment:

Table A-6: Value column values don't always line up.

Resolution: Accept.

Notes: Actioned in document.

101

Comment:

Table A-6: PRODUCT_DETAIL_PROTOCOL_CONVERTER: hyphenate and de-capitalize the first 'n' in "non-DMX".

Resolution: Accept.

Notes: Actioned in document.

102

Comment:

Table C-2 (Slot Label ID): Add 0x8000-0xFFDF (or whatever ending value is deemed appropriate) for manufacturer definitions. None of what I do matches any values in the table, other than "SD_UNDEFINED". For example, there's no "SD_FLASH_RATE" for an LED (SD_FIXTURE_SPEED is intended for all parameters, and that's not my use case). Another example, there's no "SD_PALETTE_SELECTION" for choosing from one of a set of colour palettes. There's also no "SD_PWM_RATE", "SD_WAVE_OFFSET_PARAMETER" for control of some effects, nor "SD_ADC_RESOLUTION" for control of the ADC resolution when measuring analog inputs for other types of control. My point is that the current selections in the table are inadequate for almost all my use cases and we could use a manufacturer block.

Notes: As a result of discussion on this, changed description of SD_UNDEFINED to “Not defined in this standard”. Also changed 10.6.4 last para to reference SD_UNDEFINED.

Resolution: Reject.

Notes: The point is very valid but this should be fixed in E1.37-5.

103

Comment:

Table A-6 (Product Detail Defines): It would be useful to add an "Input" section, where the product is meant to convey information back to the controller, for example, if a product exists as a temperature sensor. I'd suggest adding "PRODUCT_DETAIL_INPUT" as 0x0B00, and then we could add a few more, such as PRODUCT_DETAIL_SENSOR=0x0B01, etc. We obviously can't include everything, so I'm going to suggest adding at least the first, if not the second too.

Resolution: Accept.

Notes: Added new section with two enumerated values.

104

Comment:

6.3.2 ACK_OVERFLOW: It says, "The responder shall abort a partial transfer of overflow data for a PID when receiving a command for a different PID before the overflow data transfer is complete." In theory, this allows for a different command for the same PID. For example: GET PIDx -> ACK_OVERFLOW, then SET PIDx, then GET PIDx.

In other words, the responder could hold onto the overflow data between the two GET calls for the same PID.

Unless this behaviour is desired, I suggest this change: "The responder shall abort a partial transfer of overflow data for a PID when receiving a command for a different PID, or a different command for the same PID, before the overflow data transfer is complete." Otherwise, the text does not disallow this case. This may be desired, however.

Notes: This was not desired behaviour and the group thought the change was an improvement.

Resolution: Accept.

Notes: Actioned in document.

105

Comment:

Remove space after colon for all "GET:" and "SET:" prefixes. There appears to be both styles.

Notes: There were only a couple of instances of a space after colon.

Resolution: Accept.

Notes: Actioned in document.

106

Comment:

10.7.2 (SENSOR_VALUE): New text for "The Parameter Data in the response to a SET_COMMAND" paragraph, replace "is undefined" with:
The Parameter Data in the response to a SET_COMMAND to SENSOR_ALL_CALL and/or SUB_DEVICE_ALL_CALL shall include the sensor number requested (including SENSOR_ALL_CALL) and a value of 0x0000 for Present Value, Lowest Detected Value, Highest Detected Value, and Recorded Value.

Resolution: Accept.

Notes: Actioned in document.

107

Comment:

10.3.6 (QUEUED_MESSAGE_SENSOR_SUBSCRIBE): SET command: Add the word "(optional)" at the end of the "Packed List" item. Also, do not change the PDL from 0x01 to 0x03.
Resolution: Accept.

Notes: Actioned in document.

108

Comment:

Should "responder" be capitalised throughout the document?

Notes: There were two places with a gratuitous ‘R’. Changed to ‘r’.

Resolution: Accept.

Notes: Actioned in document.

Summary of Public Review Responses on BSR E1.20, Entertainment Technology--RDM--Remote Device Management Over DMX512 Networks

Referenced document: BSR E1.20, Entertainment Technology--RDM--Remote Device Management Over DMX512 Networks (document number CP/2018-1025r3)

ANSI public review period: 14 May through 28 June 2021

Question: What is your opinion of BSR E1.20, Entertainment Technology--RDM--Remote Device Management Over DMX512 Networks? Is it good as it is, is it good but would be better with some changes, or is it unacceptable? Please click the appropriate radio button.

Responses:

Person	Representing	It's good.	It's good but could be better.	It's unacceptable.	Comments only, no radio button response
Larry Dew	W.A. Benjamin			X	
Thierry Dupont	Robert Juliat		X		
Wayne Howell	Artistic Licence				X
Peter Newman	Open Lighting Project		X		
Peter Willis	Howard Eaton Lighting Ltd.			X	

Specific comments:

#	Commenter	Relevant clause	Comment
1	Dew		Why can't have a PID SUPPORTED_VERSION, that returns the version of the E1.20 specification that the Responder supports with the GET command and the SET would tell the Responder what version of the E1.20 specification it supports. If the Responder never gets a SET Command it assumes E1.20-2015 and does not respond with Unicode or ACK_TIMER_HI_RES. If the Controller receives an NACK Unknown_Pid response type to the GET command it can assume the Responder supports only E1.20-2015 and ALL text will be ASCII. This method allows the specification to be modified in the future by updating the SET/GET commands to refer the latest released version of E1.20-XXXX. The Parameter Data would contain a single integer i.e. 2021 which indicates the specification version that is supported. It is defined that if a device supports the latest version it MUST support ALL previous versions.
2	Newman		Given NACK has a few meanings within the standard, can I suggest replacing "with a NACK with" to "with a NACK Response with". I see four instances in the current PDF. Or possibly the more verbose "with a NACK Reason Response with", but there seems to be slightly less consistency around that full term.
3	Dew		Since the number of Sub-devices has been expanded to 65520 all of the other related specifications will need to be updated as they all list the maximum Sub-Devices as 512. E1.37-5, E1.37-7, E1.37-1 and E1.37-2.

#	Commenter	Relevant clause	Comment
			Additionally, All of the PIDs defined in E1.37-5, E1.37.7, E1.37-1 and E1.37-2 need to have their respective PID tables modified to define if "Packing Disallowed" as well if the PID is required as "Required Sub-device" Note: Many PIDs require more than just an Index value with the GET command. i.e. E1.37-5, SLOT_DETAIL requires 2 parameter, "Slot # Requested" and "Number of Details". Therefore they can not be used with the Packing PIDs.
4	Newman	previous comment 1	Was there some sort of tracked changes/diff for this round of public review, or will that only start happening with new documents/future cycles (I can appreciate integrating it into stuff that's already in progress may be a challenge). I just wanted to check I haven't missed it somewhere. [A diff file was sent to him by TSP staff.]
5	Newman	previous comment 4	"NOTE: A NACK Reason Code of NR_FORMAT_ERROR is allowed in discovery as defined in Table 6-7." versus "Responders shall not respond to valid DISC_MUTE and DISC_UN_MUTE requests with a NACK. Responders may respond to invalid DISCOVERY_COMMAND requests, excluding DISC_UNIQUE_BRANCH, with a NACK." is it only NR_FORMAT_ERROR, or any NACK?
6	Newman	previous comment 16	So just to clarify on this comment, should e.g. the device model ID be different when a fixture is in bootloader mode or when it's been set back to factory defaults? If the dimmer module subdevice I query has just been reset at the point I cache that set of supported PIDs for all its friends, how am I supposed to know it supports many more PIDs once it's finished loading?
7	Newman	previous comment 18	Just to pick back up on my array example, personally I'd consider a "variable length field" to be one that varies, like a string, whereas in the case of the "manufacturer that returns a PDL size of 24, with a UINT8, to represent 3 bytes of data.", it would be good to cover that too with a word like repeated/arrayed/multiple. I see Table A-15 mentions "Data is an array of unsigned bytes", but that's the only place in the entire standard the word array is used, also including it (or something equivalent) in the main 10.4.2 section would probably help.
8	Newman	previous comment 19	The resolved text is "This field has no meaning for a Data Type of DS_BIT_FIELD or DS_STRING. <SNIP> This field has no meaning for non-integer Data Types.". It would probably be clearer if at least the two non-integer references were brought together sentence wise, or probably combined into one sentence.
9	Newman	previous comment 20	Sorry, I think this one needs reopening, as Table A-15 now has UINT/INT64 which are not "non-integer Data Types.", but won't fit in the existing 32 bit field.
10	Newman	previous comment 23	Fix the typo "For details of PACKED_PID_INDEX see Section 10.4.7. For details of PACKED_PID_SUB see Section 10.4.7." PACKED_PID_SUB is 10.4.6, but probably wants to be first.
11	Newman	previous comment 38	Add "Section 10.4.7: Get/Set Packed List of PIDs by Index (PACKED_PID_INDEX) - Type: Improvement First Item Requested / Encoded: Add a comment similar to PACKED_PID_SUB regarding alignment for e.g. DMX_PERSONALITY_DESCRIPTION “When the index field of the PID is less than 16-bits it shall be right justified.”

#	Commenter	Relevant clause	Comment
			Also do this for "Item x:" too.
12	Newman	previous comment 40	It would be good to add a note explicitly clarifying that requests may be non-contiguous, the TG seemed to agree when they reworked it, but the only mention is still the rather subtle "(For Example: A PID of NACK_DESCRIPTION means the range is the number of defined NACK Reason Codes). A value of 0xFFFF shall require the responder to return all available data."
13	Newman	previous comment 52	Since the update linked to this comment, the standard no-longer gives any detail of what should happen to Lowest/Highest/Recorded during a set if they are supported, which seems a step backwards to me. It should at least clarify what gets reset during a reset and to what values (i.e. it's not just a reboot of the sensor). IMHO the previous text was fine, it just needed a clarification that it only applied if the respective field was supported.
14	Newman	previous comment 70	Both the comments and old comments don't appear to apply to this comment (which was about rate of data transfer), the old comments appear to be refering to comment 71. Given some of my more esoteric SI units got approved, it seems a bit odd this one got rejected, hence I wonder if it's an error. TLDR: Please add a bytes or bits per second, the equivalent Table A-12 reference and extend the existing reference about 1000/1024 ""When a prefix is used with MEMORY, the multiplier refers to binary multiple. i.e. KILO means multiply by 1024.
15	Newman	previous comment 71	Some clarification and a far better example of my previous comment. Consider either a moving head with a couple of filters which need changing/cleaning every 1,000 or 10,000 hours of operation (and a sensor to indicate this run time, which may be independent of the overall fixture or lamp hours). Or a fog machine where it's desirable to implement a ""real world"" sensor value of estimate run time (like cars do now). Currently I have to use the unit of seconds for both, for the filters, I have to use 3,600 kiloseconds or 36 megaseconds respecitively, which isn't very catchy. For the smoke machine, I might report up to 14400 decaseconds, but when it says 1080 decaseconds, do you know if it needs refilling before or after the show? For all the above scenarios, just being able to see 1,000/10,000/40 or 3 hours respectively is a lot easier to understand, and is both at a sufficient level of precision of the usage, and likewise probably an acceptable level of precision given the accuracy of the measurement.
16	Newman	previous comment 82	Another commented previously said ""DS_MAC" in Table A-16 should be renamed to "DS_ETHERNET_MAC" because there are different MACs for different transport media.". However this is confusing, as the same MAC address system is used for WiFi too, which is a different standard (hence not Ethernet). We should use the same terminology as in E1.37-2 which was INTERFACE_HARDWARE_ADDRESS_TYPE1 AKA the EUI-48 [EUI] hardware address of an interface. Or some improved compromise of that.
17	Newman	previous comment 94	Sorry to be pedantic, but ""The Slot Index is a reference to the offset from the DMX512 Starting Address"", so in Appendix C, please replace the remainder of ""Indexes"", ""Manufacturer-specific IDs"" and ""Publicly defined IDs"" with ""Types"" or ""Slot Label IDs"" as well as dropping the Index from ""Slot Index Type"", so the names

#	Commenter	Relevant clause	Comment
			used here match the stuff in SLOT_INFO.
18	Howell	?	RDM has now become very widely accepted. The current complaint is that Manufacturer Specific Pids often need a tool from that manufacturer to us. Perhaps the most important type is the 'combo box' pids where a range of numbers need a text enumeration to be meaningful. Examples of standard PIDs are PERSONALITY and PERSONALITY_DESCRIPTION. In principle e1-37-5 PARAMETER_METADATA can solve this, but for light weight devices it will be cumbersome. A controller needs to download the entire meta data file in order to tell the user what decimal 234 means. If interleaving zero start code DMX, that would be very slow. PERSONALITY and PERSONALITY_DESCRIPTION is a more elegant solution as the text for decimal 234 can be retrieved in one command. I'm calling PERSONALITY_DESCRIPTION a 'helper pid' for PERSONALITY. I would like to include the concept of a helper pid in PARAMETER_DESCRIPTION. We have an unused field Type. This can be changed to an offset from the core pid for helper pid. Example: ManSpec PID = 0x8000 PERSONALITY_DESCRIPTION.Type = 0x7f. Helper PID = 0x8000 + 0x7f = 0x807f. This would make a huge difference to support for manufacturer specific pids.
19	Willis	none	The JSON schema shown in the draft E137-5 document belongs HERE, in E1.20 NOT in E137-5. Particularly as you are wishing to deprecate the use of PARAMETER_DESCRIPTION as described in this document. It is daft to update this document without any mention that the existing PARAMETER_DESCRIPTION PID is so broken, and currently not good for much more than returning the text description of the PID.
20	Willis	6.2.7.1 PortID	The term "Controller Port" is mentioned here (and in 3.1.2), but not clearly defined. (Not in appendix D). A number of legacy controllers have incorrectly set this field to zero, and the revised standard should explicitly define what is expected of a responder in this case. Existing responders could have legitimately rejected such traffic, but market forces have required us to be lax, even though is has been the controllers at fault. The term "command port" is already used in the standard to refer to physical ports. I believe the original intent of this field was to embed in the data stream an indication of which physical port the data originated from. There are occasions where this is admirable and useful. Equally there are many cases (for example a simple splitter that does not have a processor generating new data for each physical output) where this is not possible. The standard should reflect this, and state that controllers shall set the field in the range 1-255, but that

#	Commenter	Relevant clause	Comment
			responders shall interpret 0 as a valid value and treat it as if it were 1.
21	Dew	6.2.7.1	I thought they were going to allow the PORT ID field to be 0-255, still listed as 1-255?
22	Dew	6.2.8.1	What happens if a Responder receives a Controller Flag that contains a non-supported bit as defined in table 6-3a? Should the message be rejected (Data Range Error/Format Error) or ignored with no response or ignored with a normal response? If the payload of a controller generated message contains the Unicode bit and the Responder does not support Unicode, should the responder respond with a NACK reason or ignore the bit and respond normally? What NACK reason? Section 6.2.8.1: If the payload of a controller generated message contains the Unicode bit and the response does not contain a text field, should the responder reject the message with a NACK reason or ignore the bit in the Controller Flag. If rejected what is the NACK reason?
23	Willis	6.2.8.1	Controller Flags This section needs to define what a responder should do on receipt of non-supported bits being set. This section needs to define what a responder should do on the Unicode bits being set and the responder not supporting Unicode. This section needs to define what a responder should do on the Unicode bits being set and the responder response not containing a text field. Reject with NACK – if so which one, ignore with no response ? I suggest there is an NACK: UNSUPPORTED_CONTROLLER_FLAG reason added. However I would also recommend a serious appraisal of the suggestions by others to the use of a PID SUPPORTED_PROTOCOL_VERSION. A GET would allow a controller to ascertain what the latest version of the standard a Responder is capable of, and a SET would inform the responder as to the controller capability.
24	Dew	6.3.5	If a responder supports ACK_TIMER_HI_RES why does it need to tell the Controller and therefore why does the Controller need to give the Responder permission to use it? If the Controller receives a response type of ACK_TIMER_HI_RES then the Controller should properly compute the delay time just as it did for the normal ACK_TIMER. Passing the HI_RES bit back and forth in the Control Flag seems redundant.
25	Howell	6.3.5	(RESPONSE_TYPE_NACK_REASON)" Change: "The format of the MDB of the response message is defined as follows:"

#	Commenter	Relevant clause	Comment
			To: "The format of the MDB of the response message is shown in the following example:" This way, the text doesn't imply fixing the CC to GET_COMMAND_RESPONSE.
26	Howell	6.3.5	(RESPONSE_TYPE_NACK_REASON)" The RESPONSE_TYPE_NACK_REASON shows the response as using GET_COMMAND_RESPONSE. This could be just an example, or it could mean to always use GET_COMMAND_RESPONSE, even for SET_COMMAND requests. If it's always supposed to be GET_COMMAND_RESPONSE, then I think we should add a sentence: "A GET_COMMAND_RESPONSE shall always be used as the Command Class." If this is not true, I would change the "(CC)" table element to "GET_COMMAND_RESPONSE or SET_COMMAND_RESPONSE".
27	Willis	7.5	Typo – should read Discovery Unique Branch
28	Willis	9	The example in this section refers to dimmer racks, the use of which is becoming increasingly rare with the advent of LED technology and the demise of incandescent fixtures. The use case for increasing the number of sub-devices is not given in this revision. The example should be revised or added to in order to make the document relevant to the likely use case. I am firmly of the view that allowing 65320 sub-devices is the wrong decision. Since there is no way of restricting this to RDMNet applications, the existence of so many sub-devices at the RS485 level could completely swamp the available bandwidth. A legacy controller should be capable of operating with a responder compliant with this new standard, albeit without the benefit of any new features. Thus it should be possible to discover and set the address of any fixture. A controller written to the existing standard could legitimately reject a DEVICE_INFO declaring more than 512 sub-devices, and thus not be able to control or configure the basics of a target device. Being able to handle this many sub-devices puts an unnecessary burden on controllers.
29	Dew	9.2.1	Why is there a need for 65320 Sub-Devices? *[sic. The clause gives a range of 65520 device.]*
30	Willis	9.2.2	The first and second paragraphs are confusing, as they discuss both the number of sub-devices and the means of getting the list of supported parameters. I suggest moving para1 sentence 2 to become a third paragraph – i.e. after the discussion about how many there are and how they find need to be scanned.
31	Willis	9.2.3	This section does not mention a number of required Messages (such as DEVICE_INFO) mentioned in table A-3.

#	Commenter	Relevant clause	Comment
			The section should at least reference Table A-3, as this table has been enhanced since the earlier revision of the standard.
32	Dew	10.1.1	In the case of a Unicode response the Responder could use the Response Type of the ACK_UNICODE to indicate the text field is Unicode, otherwise the normal ACK indicates the text field would be ASCII. If the Controller does not support E1.20-20xx then the responder MUST respond with ASCII text and an ACK conforming to the E1.20-2015 specification. By not changing the meaning of the Message Count, field there should be minimum changes to Controller/Responder code. Since we only need to add ACK_UNICODE as a new Response Type (we already added ACK_TIMER_HI_RES) we need minimal changes in the Controller and Responder code. ACK_UNICODE could be used equivalent to ACK even if no text fields are present OR the ACK_UNICODE could be restricted to only responses that contain text fields. This method keeps the existing structure of the E1.20 specification intact by NOT adding special meaning bits within the message structure but by using PIDs to interface with the Controllers and Responders. Keeping in mind that RdmNet relies on the E1.20 specification changing the Message structure could cause adverse issues and by using PIDs and Response Types to add functionality is more keeping with the intent of the original E1.20 specification and subsequent E1.37-X add on specifications.
33	Willis	10.3.1	This section uses the term “subscribed sensor”, but there is no mention elsewhere in the document as to what this is. The paragraph under the PID example that reads “The status type requested is …. Or as allowed in table 10-1” is still confusing, as Table 10-1 is a table of required responses. The correct reference should be Table A-4.
34	Willis	10.3.2.2	The note about the earlier standard not allowing use of STATUS_NONE should really be in 10.3.1 as it relates to the outgoing GET. Indeed the use of STATUS_NONE is to be preferred if you just want to poll for Queued Messages and not be swamped by possible error, advisory or warning messages.
35	Willis	10.3.6	QUEUED_MESSAGE_SENSOR_SUBSCRIBE Incorrectly referenced in table A-3?
36	Dew	10.3.6	The section 10.3.6 defines the pid as QUEUED_MESSAGE_SENSOR_SUBSCRIBE, however in table A-3 it is defined as QUEUED_MESSAGE_PID_SUBSCRIBE. I assume it should be as defined in 10.3.6
37	Willis	10.4.3	Get supported Parameters Enhanced PID Support - incorrect forward reference for PACKED_PID_SUB, which is 10.4.6 not 10.4.7. Consider referencing these in the order in which they appear later in the document. (ie SUB, then INDEX)
38	Willis	10.4.3	There is an issue with table numbering – why is there a table 10-1a?

#	Commenter	Relevant clause	Comment
39	Newman	10.4.3	Is the "Non-identical Sub-Device" flag actually a per-subdevice rather than per PID flag? i.e. if it's set to 1 on any PID, should it be set to 1 for all of them? If so, why is it repeated multiple times in the data, not once as a non-repeating set of flags at the start of the data? The comment below Table 10-1a seems to imply it would be set to true for DMX_PERSONALITY for example if each subdevice had a different set of personalities available (for example, which doesn't seem to be disallowed by the standard). Whereas the descriptions above the table "determine if all sub-devices have the same list of supported parameters" and "If the "Non-identical Sub-Device" flag is not set for any parameter in a Supported Parameters Enhanced response, then each Sub-Device shall have an identical response for SUPPORTED_PARAMETERS_ENHANCED to each other Sub-Device." all appear, by my reading at least, to be related to the PID list in SUPPORTED_PARAMETERS_ENHANCED rather than what you get when you query the listed PID.
40	Willis	10.4.5	Get Nack Reason Description This is missing a corresponding PID to get a list of valid manufacturer specific NACK reason codes. I don’t want to have to poll all possible values to know which ones are valid and which ones are erroneous.
41	Dew	10.4.5	I have no idea how to know what Custom NACK Reason to ask for by number using the new PID NACK_DESCRIPTION. I can't find anywhere the device tells me what the custom numbers are. Do I what until I receive a NACK with a reason greater than 0x8000 and then ask for the description? How do I know it is not a mistake reason? It seems there should be a PID similar to SUPPORTED_PARAMETERS, called MANUFACTURER_NACK_REASONS that returns a list of manufacturer defined NACK Reasons. Then just like the PID PARAMETER_DRECRIPTION the Responder can be queried for the descriptions using NACK_DESCRIPTION from the list returned
42	Willis	10.4.6	Get Packed List for Sub-Devices I found the terminology in this section confusing, as to when “sub-device means sub-device not including root, and all devices, ie including root.” Elsewhere we refer to the Root device, and Sub-Devices. All of a sudden in this section, you are using the term Sub-Device to include the root device. Consider changing the field names referring to First Sub-devices, Number of Sub-devices, to First Device, and Number of Devices. The text for First Device Requested would then read “ The device encoded for entry x of the response packet. A value of 0x0000 indicates the root device. Values in the range 0x0001-0xFFF0 indicate a sub-device. It would then be clearer why the Number of Devices is in the range “0x0001 to 0xFFF1 (one more than the number of sub-devices described elsewhere in the document as 0xFFF0)

#	Commenter	Relevant clause	Comment
43	Willis	10.4.6	Get Packed List for Sub-Devices The example GET:PID shows the number of sub-devices requested as 0xFFFF, but the earlier section (top of page 71) states that this field shall be in the range 0x0001-0xFFF1. By my reckoning the example should have 0x0003 in this field.
44	Newman	10.4.6	Can a responder implement this PID while still behaving as per the previous version of this standard with non-contiguous sub device numbers? If no, please explicitly clarify that. If it can, say I have sub devices at 0x0001, 0x0003 and 0x0005, and I do a PACKED_PID_SUB for the first as 0x0001 and request three sub devices, does the whole thing NACK due to "if any of the packed replies would generate a NACK response, the responder shall return a NACK Response" which would be triggered by my sub device out of range, or does it return the intervening devices with a PDL of 0, or does it only return the ones that exist? A clarifiying note would be helpful in all regards. If it's only supposed to handle contiguous ones, the encoding of all item values seems a bit redundant. If non-contiguous is allowed, having an example showing this too would be very beneficial.
45	Newman	10.4.7	If you consider say GET SELF_TEST_DESCRIPTION, which doesn't explicitly require contiguous self test numbers, so say I have self tests at 0x01, 0x03 and 0x05, and I do a PACKED_PID_INDEX for the first as 0x01 and request three items, does the whole thing NACK due to "if any of the packed replies would generate a NACK response, the responder shall return a NACK Response" which would be triggered by my data out of range, or does it return the intervening numbers with a PDL of 0, or does it only return the ones that exist? A clarifiying note would be helpful in all regards. If the latter, I don't know if there is some way the PD table can better represent this (the sub device one is clearer, if it had item for entry x/x+1 etc). If it's only supposed to handle contiguous ones, the encoding of all item values seems a bit redundant.
46	Dupont	10.6.4	When we wanted to implement them we noticed that several controller manufacturers did not understand what the standard requires. It is not easy to understand but easy to interpret. You have to go throug a long video on Ethernet about this subject to get some clarifications. It is all due to a lack of a clear list of examples that we would like to propose. Those examples are from a real implementation and work with many well known controllers The following example is from our latest product SULLY. Slot Info is currently implemented and works with several lighting consoles from leading manufacturers. We wanted to show the issue as RDM Integrity shows it as a lot of manufacturers are using this test tool. Unfortunately RDM Integrity is wrong when you have a 16 bit slot and it creates a lot of issues. Even for us it is difficult to explain clearly what to do. The standard in this occasion uses words and table that are misleading and many controllers fell into the trap. The issue is often in 16 bit. Our example: Mode 6: Followspot16b

#	Commenter	Relevant clause	Comment
			Dimmer Dimmer fine Master Master fine Strobe duration Strobe speed Response time Control mode Slot Offset [0x0000, (0)] Slot Type [0x00, (0)] Primary Slot Slot Label ID [0x0001, (1)], Intensity --- Slot Offset [0x0001, (1)] Slot Type [0x01, (1)] Secondary Fine RDM Integrity is wrong by showing this : Slot Label ID [0x0000, (0)], Slot Label ID Undefined Entry [0x0000, (0)] It is a secondary slot, so the Slot Label ID should be a reference to the offset of the slot that is its primary. --- Slot Offset [0x0002, (2)] Slot Type [0x00, (0)] Primary Slot Slot Label ID [0x0002, (2)], Intensity Master --- Slot Offset [0x0003, (3)] Slot Type [0x01, (1)] Secondary Fine RDM integrity is wrong: Slot Label ID [0x0002, (2)], Intensity Master --- Slot Offset [0x0004, (4)] Slot Type [0x00, (0)] Primary Slot Slot Label ID [0x0404, (1028)], Strobe --- Slot Offset [0x0005, (5)] Slot Type [0x00, (0)] Primary Slot Slot Label ID [0x0404, (1028)], Strobe --- Slot Offset [0x0006, (6)] Slot Type [0x00, (0)] Primary Slot Slot Label ID [0x0503, (1283)], Fixture Speed --- Slot Offset [0x0007, (7)] Slot Type [0x00, (0)] Primary Slot

#	Commenter	Relevant clause	Comment
			Slot Label ID [0x0502, (1282)], Fixture Control
47	Dupont	10.6.5	When we wanted to implement them we noticed that several controller manufacturers did not understand what the standard requires. It is not easy to understand but easy to interpret. You have to go throug a long video on Ethernet about this subject to get some clarifications. It is all due to a lack of a clear list of examples that we would like to propose. Those examples are from a real implementation and work with many well known controllers The following example is from our latest product SULLY. Slot Info is currently implemented and works with several lighting consoles from leading manufacturers. We wanted to show the issue as RDM Integrity shows it as a lot of manufacturers are using this test tool. Unfortunately RDM Integrity is wrong when you have a 16 bit slot and it creates a lot of issues. Even for us it is difficult to explain clearly what to do. The standard in this occasion uses words and table that are misleading and many controllers fell into the trap. The issue is often in 16 bit. Our example: Mode 6: Followspot16b Dimmer Dimmer fine Master Master fine Strobe duration Strobe speed Response time Control mode Slot Offset [0x0000, (0)] Slot Type [0x00, (0)] Primary Slot Slot Label ID [0x0001, (1)], Intensity ---------- Slot Offset [0x0001, (1)] Slot Type [0x01, (1)] Secondary Fine RDM Integrity is wrong by showing this : Slot Label ID [0x0000, (0)], Slot Label ID Undefined Entry [0x0000, (0)] It is a secondary slot, so the Slot Label ID should be a reference to the offset of the slot that is its primary. ---------- Slot Offset [0x0002, (2)] Slot Type [0x00, (0)] Primary Slot Slot Label ID [0x0002, (2)], Intensity Master ---------- Slot Offset [0x0003, (3)] Slot Type [0x01, (1)] Secondary Fine RDM integrity is wrong: Slot Label ID [0x0002, (2)], Intensity Master

#	Commenter	Relevant clause	Comment
			---------- Slot Offset [0x0004, (4)] Slot Type [0x00, (0)] Primary Slot Slot Label ID [0x0404, (1028)], Strobe ---------- Slot Offset [0x0005, (5)] Slot Type [0x00, (0)] Primary Slot Slot Label ID [0x0404, (1028)], Strobe ---------- Slot Offset [0x0006, (6)] Slot Type [0x00, (0)] Primary Slot Slot Label ID [0x0503, (1283)], Fixture Speed ---------- Slot Offset [0x0007, (7)] Slot Type [0x00, (0)] Primary Slot Slot Label ID [0x0502, (1282)], Fixture Control
48	Dupont	10.11.4	Get/Set Perform Self Test (PERFORM_SELFTEST)SELFTEST/ The standard says that you may return pass/fail status or other information by queuing a Status Message response. We implemented this but the reality is that very few controllers display Status_Message especially the major brands (lighting consoles for example). - We created a Manufacturer Pid to display the result. The issue with this solution is that the status has to be seen in other screen or page and not right next to Selftest PID. - Would it be possible to create an official PID (Value: 0x1022) to display the pass/fail status? We found that the controller has to check wether the 254 potential tests are present or not and it is relatively long. - Would it be possible to declare the number of tests available in the description or returning something to tell the controller that there is no more selftests (end of enumeration)?
49	Willis	11.1	Polling Intervals Para 2 – the Note should be on a new line. What is the justification for restricting polling to intervals of 5 seconds? Either justify the time interval or remove the advice.
50	Willis	Appendix A Table A-3	The PID value for SUPPORTED_PARAMETERS_ENHANCED is incorrect
51	Dew	Appendix A Table A-3	The section 10.3.6 defines the pid as QUEUED_MESSAGE_SENSOR_SUBSCRIBE, however in table A-3 it is defined as QUEUED_MESSAGE_PID_SUBSCRIBE. I assume it should be as defined in 10.3.6
52	Dew	Appendix A	Page 120 Support Parameters Enhanced has the same PID # as Supported Parameters

#	Commenter	Relevant clause	Comment
		Table A-3	
53	Willis	Appendix A Table A-3	Lamp ON Mode formatting issue
54	Willis	Appendix A Table A-3	PIDS 0xFFE0-0xFFFF – please drop the reference to “Future”. These are already in use for development purposes.
55	Newman	Appendix A Table A-4	"STATUS_GET_LAST_ MESSAGE" has an extra space after the last underscore. In general, running a search for "_<space>" would be useful before publication (although it may throw up the odd red herring in tables where things wrap), as I don't think this is the first one I've reported.
56	Willis	Appendix A Table A-12	Missing text/description for entry 0x23??
57	Dew	Appendix A Table A-12	0x23 has NO text
58	Newman	Appendix A Table A-12	Remove 0x23 with no definition.
59	Newman	Appendix A Table A-14	Negative SI multipliers for bits and other integer units, it would be good to have a note flagging these as discouraged or disallowed, my half joking suggestion of bits is the only way you could have less than a byte. Probably likewise for UNITS_NONE, if it was some sort of counter it shouldn't have a fraction. See related https://www.rdmprotocol.org/forums/showthread.php?t=1326
60	Newman	Appendix A Table A-14	The note about binary mulitples and memory, should it be changed to SENS_MEMORY as per A-12. Or perhaps better yet, rewrite all those notes to be along the lines of "When a prefix is used with UNITS_BYTE, the multiplier..." and similar for the other notes. See related https://www.rdmprotocol.org/forums/showthread.php?t=1326
61	Newman	Appendix A Table A-15	Given some recent comments, is it worth adding nominative references to the table for URL/MAC/IPv4/IPv6 (as already present in other docs) to avoid any confusion about which MAC is being referred to?
62	Howell	Appendix D	We need to tighten our definition of relationship between prefixes and digital data types. Eg. Kilo = 1000 vs 1024.
63	Howell	Appendix D	Add a definition for a port. Port: A means of physical connection of an RDM signal to a device.

Peter Newman, via email on 18 June 2021, wrote, “I've got a couple of more "procedural" queries arising from the resolutions: . . . .” The “queries” were about previous comments 1 and 70, and also were offered on Newman's current public review response form. They are listed above.