BSR/ASHRAE Addendum *cq* to
ANSI/ASHRAE Standard 135-2020

Public Review Draft

Proposed Addendum *cq* to Standard 135-2020, BACnet® - A Data Communication Protocol for Building Automation and Control Networks

First Public Review (September 2023)
(Draft shows Proposed Changes to Current Standard)

This draft has been recommended for public review by the responsible project committee. To submit a comment on this proposed standard, go to the ASHRAE website at www.ashrae.org/standards-research--technology/public-review-drafts and access the online comment database. The draft is subject to modification until it is approved for publication by the Board of Directors and ANSI. Until this time, the current edition of the standard (as modified by any published addenda on the ASHRAE website) remains in effect. The current edition of any standard may be purchased from the ASHRAE Online Store at www.ashrae.org/bookstore or by calling 404-636-8400 or 1-800-727-4723 (for orders in the U.S. or Canada).

This standard is under continuous maintenance. To propose a change to the current standard, use the change submittal form available on the ASHRAE website, www.ashrae.org.

The appearance of any technical data or editorial material in this public review document does not constitute endorsement, warranty, or guaranty by ASHARE of any product, service, process, procedure, or design, and ASHRAE expressly disclaims such.

ASHRAE, 180 Technology Parkway NW, Peachtree Corners, GA 30092

[This foreword, the table of contents, the introduction, and the "rationales" on the following pages are not part of this standard. They are merely informative and do not contain requirements necessary for conformance to the standard.]

FOREWORD

The purpose of this addendum is to present a proposed change for public review. These modifications are the result of change proposals made pursuant to the ASHRAE continuous maintenance procedures and of deliberations within Standing Standard Project Committee 135. The proposed changes are summarized below.

In the following document, language to be added to existing clauses of ANSI/ASHRAE 135-2020 is indicated through the use of *italics*, while deletions are indicated by ~~strikethrough~~. Where entirely new subclauses are proposed to be added, plain type is used throughout. Only this new and deleted text is open to comment at this time. All other material in this document is provided for context only and is not open for public review comment except as it relates to the proposed changes.

The use of placeholders like XX, YY, ZZ, X1, X2, NN, x, n, ? etc., should not be interpreted as literal values of the final published version. These placeholders will be assigned actual numbers/letters only after final publication approval of the addendum.

135-2020*cq*-1 Define a new "short form" for Array, List, and SequenceOf base types.

Rationale

The requirement by ANNEX Z to use a JSON object for every structured base type defined in ANNEX Y can become verbose for base types Array, List, and SequenceOf. In many cases, the position indicators can be inferred, and therefore the member names of "1", "2", "3", etc., are unnecessary and undesirable.

[Add new **Clause Z.3.X**, p. 1369]

Z.3.X Short Form for Array, List, and SequenceOf

If all values of an Array, List, or SequenceOf are present in the representation, then the values can be represented as a JSON array, with the numerical indexes implied.

For example, a representation in the form

"foo": { "$base":"List", "$"1": 55, "2":66, "3":77, "4":88 }

can have the values be represented as a JSON array as:

"foo": { "$base":"List", "$values": [55, 66, 77, 88] }

Note that the JSON member name is "$values" rather than the scalar "$value" that is used for primitive base types.

If all values are present and no metadata is present, then the "short form" encoding can be used. In this case, the values are represented directly as a JSON array.

For example, a representation in the form

"foo": { "$values": [55, 66, 77, 88] }

can be represented in the short form as

"foo": [55, 66, 77, 88]

This applies to non-primitive members as well, so

"foo": { "$values": [{"a":55, "b":66 } , {"a":77, "b":88 }] }

can be represented in the short form as

"foo": [{"a":55, "b":66 }, {"a":77, "b":88 }]

The short form cannot be used when metadata is present. This is often the case when the definition of the Array, List, or SequenceOf is not determined by its context and therefore must convey metadata for $base, $type, or $memberType. For example, the following cannot use the short form:

"foo": { "$base":"List", "$memberType":"Real", "$values": [55, 66, 77, 88] },
"bar": { "$type":"555-BarType", "$values": [55, 66, 77, 88] }

The JSON array form cannot be used when a representation does not include all the values of the Array, List, or SequenceOf and therefore the position numbers cannot be inferred to be "1", "2", "3", etc. For example, the following representation cannot use the JSON array form.

"big-list": { "4231": 55, "4232":66, "4233":77, "4234":88 }

Note that the JSON array form for Array, List, and SequenceOf was added in Protocol_Revision N and there may be consumers that only parse the original long form with numeric member names. Therefore, if it is not known to the generator that the consumer can parse the short form, the numeric form should be used. For dynamic generation, this might be determined by communication, e.g., BACnet/WS. For file generation, the knowledge of the consumer's capabilities might be determined by the use case of the file, e.g., use cases that were developed after the array form was defined.

135-2020*cq*-2 Formally define the existing "short form" for primitives.

Rationale

The "short form" of the JSON representation of a primitive data item's value was never formally defined in ANNEX Z, even though it is defined for metadata in ANNEX W and is widely used in examples of values.

[Add new **Clause Z.5.X**, p. 1369]

Z.5.X Short Form for Primitive Values

If the value is the only thing being represented for a primitive base type, then the "short form" can optionally be used. In this case, the appropriate JSON type specified in Table Z-1 is used as the value of the JSON object member.

For example, the representations

```
"my-real": { "$value": 75 }
"my-boolean": { "$value": false }
"my-string": { "$value": "Hello"}
```

Can be represented in short form as

```
"my-real": 75
"my-boolean": false
"my-string": "Hello"
```

Note that the Null base type does not have a value and therefore does not have a "short form". If a Null data item has no metadata, it is represented as

```
"my-null": {}
```


[Add a new entry to **History of Revisions**, p. 1429]

(This History of Revisions is not part of this standard. It is merely informative and does not contain requirements necessary for conformance to the standard.)

HISTORY OF REVISIONS

. . .	. . .	**. . .**
1	X	**Addendum *xx* to ANSI/ASHRAE Standard 135-2020** Approved by ASHRAE on MONTH DAY, 20XX; and by the American National Standards Institute on MONTH DAY, 20XX. **1. Define a new "short form" for Array, List, and SequenceOf base types.** **2. Formally define the existing "short form" for primitives.**

