



SLOVENSKI STANDARD
oSIST prEN IEC 62541-13:2018
01-november-2018

Enotna arhitektura OPC - 13. del: Agregati

OPC Unified Architecture - Part 13: Aggregates

iTeh STANDARD PREVIEW
(standards.iteh.ai)

Architecture unifiée OPC - Partie 13: Agrégats

Ta slovenski standard je istoveten z: prEN IEC 62541-13:2018

<https://standards.iteh.ai/catalog/standards/sist/cef5cf4e-d5be-4a1f-a050-396629323059/sist-en-iec-62541-13-2020>

ICS:

25.040.40	Merjenje in krmiljenje industrijskih postopkov	Industrial process measurement and control
35.240.50	Uporabniške rešitve IT v industriji	IT applications in industry

oSIST prEN IEC 62541-13:2018

en,fr,de



65E/613/CDV

COMMITTEE DRAFT FOR VOTE (CDV)

PROJECT NUMBER: IEC 62541-13 ED2	
DATE OF CIRCULATION: 2018-08-17	CLOSING DATE FOR VOTING: 2018-11-09
SUPERSEDES DOCUMENTS: 65E/561/RR	

IEC SC 65E : DEVICES AND INTEGRATION IN ENTERPRISE SYSTEMS	
SECRETARIAT: United States of America	SECRETARY: Mr Donald (Bob) Lattimer
OF INTEREST TO THE FOLLOWING COMMITTEES:	PROPOSED HORIZONTAL STANDARD: <input type="checkbox"/> Other TC/SCs are requested to indicate their interest, if any, in this CDV to the secretary.
FUNCTIONS CONCERNED: <input type="checkbox"/> EMC <input type="checkbox"/> ENVIRONMENT <input type="checkbox"/> QUALITY ASSURANCE <input type="checkbox"/> SAFETY	
<input checked="" type="checkbox"/> SUBMITTED FOR CENELEC PARALLEL VOTING Attention IEC-CENELEC parallel voting The attention of IEC National Committees, members of CENELEC, is drawn to the fact that this Committee Draft for Vote (CDV) is submitted for parallel voting. The CENELEC members are invited to vote through the CENELEC online voting system.	<input type="checkbox"/> NOT SUBMITTED FOR CENELEC PARALLEL VOTING

This document is still under study and subject to change. It should not be used for reference purposes.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

TITLE:

OPC Unified Architecture - Part 13: Aggregates

PROPOSED STABILITY DATE: 2021

NOTE FROM TC/SC OFFICERS:

CONTENTS

FIGURES	5
TABLES	5
FOREWORD	2
1 Scope	9
2 Normative references	9
3 Terms, definitions, and abbreviations	9
3.1 Terms and definitions	9
3.2 Abbreviations	12
4 Aggregate Information Model	12
4.1 General	12
4.2 Aggregate Objects	12
4.2.1 General	12
4.2.2 AggregateFunction Object	13
4.3 MonitoredItem AggregateFilter	15
4.3.1 MonitoredItem AggregateFilter Defaults	15
4.3.2 MonitoredItem Aggregates and Bounding Values	15
4.4 Exposing Supported Functions and Capabilities	15
5 Aggregate specific usage of Services	16
5.1 General	16
5.2 Aggregate data handling	17
5.2.1 Overview	17
5.2.2 ReadProcessedDetails structure overview	17
5.2.3 AggregateFilter structure overview	17
5.3 Aggregates StatusCodes	17
5.3.1 Overview	17
5.3.2 Operation level result codes	18
5.3.3 Aggregate Information Bits	18
5.4 Aggregate details	19
5.4.1 General	19
5.4.2 Common characteristics	20
5.4.3 Specific Aggregated data handling	23
Annex A (informative) Aggregate Specific examples – Historical Access	54
A.1 Historical Aggregate specific characteristics	54
A.1.1 Example Aggregate data – Historian 1	54
A.1.2 Example Aggregate data – Historian 2	55
A.1.3 Example Aggregate data – Historian 3	56
A.1.4 Example Aggregate data – Historian 4	57
A.2 Interpolative	57
A.2.1 Description	57
A.2.2 Interpolative data	57
A.3 Average	59
A.3.1 Description	59
A.3.2 Average data	59
A.4 TimeAverage	60
A.4.1 Description	60

A.4.2	TimeAverage data	60
A.5	TimeAverage2	61
A.5.1	Description	61
A.5.2	TimeAverage2 data	61
A.6	Total	63
A.6.1	Description	63
A.6.2	Total data	63
A.7	Total2	64
A.7.1	Description	64
A.7.2	Total2 data	65
A.8	Minimum	66
A.8.1	Description	66
A.8.2	Minimum data	66
A.9	Maximum	66
A.9.1	Description	66
A.9.2	Maximum data	67
A.10	MinimumActualTime	67
A.10.1	Description	67
A.10.2	MinimumActualTime data	67
A.11	MaximumActualTime	68
A.11.1	Description	68
A.11.2	MaximumActualTime data	68
A.12	Range	68
A.12.1	Description	68
A.12.2	Range data	69
A.13	Minimum2	69
A.13.1	Description	69
A.13.2	Minimum2 data	69
A.14	Maximum2	70
A.14.1	Description	70
A.14.2	Maximum2 data	70
A.15	MinimumActualTime2	70
A.15.1	Description	70
A.15.2	MinimumActualTime2 data	71
A.16	MaximumActualTime2	71
A.16.1	Description	71
A.16.2	MaximumActualTime2 data	71
A.17	Range2	72
A.17.1	Description	72
A.17.2	Range2 data	72
A.18	AnnotationCount	73
A.18.1	Description	73
A.18.2	AnnotationCount data	73
A.19	Count	73
A.19.1	Description	73
A.19.2	Count data	73
A.20	DurationInStateZero	74
A.20.1	Description	74
A.20.2	DurationInStateZero data	74

A.21	DurationInStateNonZero	74
A.21.1	Description	74
A.21.2	DurationInStateNonZero data	74
A.22	NumberOfTransitions	75
A.22.1	Description	75
A.22.2	NumberOfTransitions data	75
A.23	Start	75
A.23.1	Description	75
A.23.2	Start data	75
A.24	End	76
A.24.1	Description	76
A.24.2	End data	76
A.25	StartBound	77
A.25.1	Description	77
A.25.2	StartBound data	77
A.26	EndBound	78
A.26.1	Description	78
A.26.2	EndBound data	78
A.27	Delta	78
A.27.1	Description	78
A.27.2	Delta data	78
A.28	DeltaBounds	79
A.28.1	Description	79
A.28.2	DeltaBounds data	79
A.29	DurationGood	80
A.29.1	Description	80
A.29.2	DurationGood data	80
A.30	DurationBad	80
A.30.1	Description	80
A.30.2	DurationBad data	81
A.31	PercentGood	81
A.31.1	Description	81
A.31.2	PercentGood data	81
A.32	PercentBad	82
A.32.1	Description	82
A.32.2	PercentBad data	82
A.33	WorstQuality	83
A.33.1	Description	83
A.33.2	WorstQuality data	83
A.34	WorstQuality2	84
A.34.1	Description	84
A.34.2	WorstQuality2 data	84
A.35	StandardDeviationSample	85
A.35.1	Description	85
A.35.2	StandardDeviationSample data	85
A.36	VarianceSample	85
A.36.1	Description	85
A.36.2	VarianceSample data	85
A.37	StandardDeviationPopulation	86

A.37.1	Description.....	86
A.37.2	StandardDeviationPopulation data	86
A.38	VariancePopulation	87
A.38.1	Description.....	87
A.38.2	VariancePopulation data	87

FIGURES

Figure 1	– Representation of Aggregate Configuration information in the AddressSpace	16
Figure 2	– Variable with Stepped = False and Simple Bounding Values.....	24
Figure 3	– Variable with Stepped = True and Interpolated Bounding Values	25

TABLES

Table 1	– Interpolation examples.....	10
Table 2	– AggregateConfigurationType Definition.....	12
Table 3	– Aggregate Functions Definition	13
Table 4	– AggregateFunctionType Definition	13
Table 5	– Standard AggregateType Nodes	14
Table 6	– ReadProcessedDetails	17
Table 7	– AggregateFilter structure	17
Table 8	– Bad operation level result codes	18
Table 9	– Uncertain operation level result codes	18
Table 10	– Data location	18
Table 11	– Additional information	19
Table 12	– History Aggregate interval information	20
Table 13	– Standard History Aggregate Data Type information.....	22
Table 14	– Aggregate table description	26
Table 15	– Interpolative Aggregate summary.....	27
Table 16	– Average Aggregate summary	28
Table 17	– TimeAverage Aggregate summary	28
Table 18	– TimeAverage2 Aggregate summary	30
Table 19	– Total Aggregate summary	30
Table 20	– Total2 Aggregate summary	31
Table 21	– Minimum Aggregate summary.....	32
Table 22	– Maximum Aggregate summary.....	32
Table 23	– MinimumActualTime Aggregate summary	33
Table 24	– MaximumActualTime Aggregate summary.....	34
Table 25	– Range Aggregate summary.....	34
Table 26	– Minimum2 Aggregate summary.....	36
Table 27	– Maximum2 Aggregate summary	36
Table 28	– MinimumActualTime2 Aggregate summary.....	37
Table 29	– MaximumActualTime2 Aggregate summary.....	38
Table 30	– Range2 Aggregate summary.....	38

Table 31 – AnnotationCount Aggregate summary	39
Table 32 – Count Aggregate summary.....	40
Table 33 – DurationInStateZero Aggregate summary	40
Table 34 – DurationInStateNonZero Aggregate Summary.....	41
Table 35 – NumberOfTransitions Aggregate summary	41
Table 36 – Start Aggregate summary	43
Table 37 – End Aggregate summary.....	43
Table 38 – Delta Aggregate summary.....	44
Table 39 – StartBound Aggregate summary	44
Table 40 – EndBound Aggregate summary.....	45
Table 41 – DeltaBounds Aggregate summary	45
Table 42 – DurationGood Aggregate summary	46
Table 43 – DurationBad Aggregate summary	47
Table 44 – PercentGood Aggregate summary	48
Table 45 – PercentBad Aggregate summary.....	48
Table 46 – WorstQuality Aggregate summary.....	49
Table 47 – WorstQuality2 Aggregate summary.....	50
Table 48 – StandardDeviationSample Aggregate summary	50
Table 49 – VarianceSample Aggregate summary	51
Table 50 – StandardDeviationPopulation Aggregate summary.....	52
Table 51 – VariancePopulation Aggregate summary.....	52

SIST EN IEC 62541-13:2020

<https://standards.iteh.ai/catalog/standards/sist/cef5cf4e-d5be-4a1f-a050-396629323059/sist-en-iec-62541-13-2020>

INTERNATIONAL ELECTROTECHNICAL COMMISSION

OPC UNIFIED ARCHITECTURE –**Part 13: Aggregates****FOREWORD**

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

The main task of IEC technical committees is to prepare International Standards. However, a technical committee may propose the publication of a technical report when it has collected data of a different kind from that which is normally published as an International Standard, for example "state of the art".

IEC 62541-13 has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation.

The text of this technical report is based on the following documents:

Enquiry draft	Report on voting
65E/XX/DTR	65E/XX/RVC

Full information on the voting for the approval of this technical report can be found in the report on voting indicated in the above table.

This third edition cancels and replaces the second edition of IEC 62541, published in 2015.

This edition includes the following significant technical changes with respect to the previous edition:

- a) No technical changes but numerous clarifications. Also some corrections to the examples.

52

53 This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

54 Throughout this document and the referenced other Parts of the series, certain document
55 conventions are used:

56 Italics are used to denote a defined term or definition that appears in the "Terms and definition"
57 clause in one of the parts of the series.

58 Italics are also used to denote the name of a service input or output parameter or the name of a
59 structure or element of a structure that are usually defined in tables.

60 The italicized terms and names are also often written in camel-case (the practice of writing
61 compound words or phrases in which the elements are joined without spaces, with each element's
62 initial letter capitalized within the compound). For example the defined term is AddressSpace instead
63 of Address Space. This makes it easier to understand that there is a single definition for
64 AddressSpace, not separate definitions for Address and Space.

65 A list of all parts of the IEC 62541 series, published under the general title *OPC Unified Architecture*,
66 can be found on the IEC website.

67 The committee has decided that the contents of this publication will remain unchanged until the
68 stability date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data related to the
69 specific publication. At this date, the publication will be

- 70 • reconfirmed,
- 71 • withdrawn,
- 72 • replaced by a revised edition, or
- 73 • amended.

74

75 The National Committees are requested to note that for this publication the stability date is 2021.

76 THIS TEXT IS INCLUDED FOR THE INFORMATION OF THE NATIONAL COMMITTEES AND WILL BE DELETED AT
77 THE PUBLICATION STAGE.

78 A bilingual version of this publication may be issued at a later date.

79

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

80

81

82

83
84
85
86
87
88

OPC Unified Architecture Specification

Part 13: Aggregates

89 1 Scope

90 This specification is part of the overall OPC Unified Architecture specification series and
91 defines the information model associated with Aggregates.

92 2 Normative references

93 The following documents, in whole or in part, are normatively referenced in this document and
94 are indispensable for its application.

95 IEC TR 62541-1, *OPC Unified Architecture – Part 1: Overview and Concepts*

96 IEC 62541-3, *OPC Unified Architecture – Part 3: Address Space Model*

97 IEC 62541-4, *OPC Unified Architecture – Part 4: Services*

98 IEC 62541-5, *OPC Unified Architecture – Part 5: Information Model*

99 IEC 62541-8, *OPC Unified Architecture – Part 8: Data Access*

100 IEC 62541-11, *OPC Unified Architecture – Part 11: Historical Access*

101 3 Terms, definitions, and abbreviations

102 3.1 Terms and definitions

103 For the purposes of this document, the terms and definitions given in IEC TR 62541-1,
104 IEC 62541-3, IEC 62541-4, and IEC 62541-11 as well as the following apply.

105 3.1.1

106 **ProcessingInterval**

107 timespan for which derived values are produced based on a specified *Aggregate*

108 Note 1 to entry: The total time domain specified for *ReadProcessed* is divided by the *ProcessingInterval*. For
109 example, performing a 10-minute *Average* over the time range 12:00 to 12:30 would result in a set of three
110 intervals of *ProcessingInterval* length, with each interval having a start time of 12:00, 12:10 and 12:20 respectively.
111 The rules used to determine the interval *Bounds* are discussed in 5.4.2.2.

112 3.1.2

113 **interpolated**

114 data that is calculated from data samples

115 Note 1 to entry: Data samples may be historical data or buffered real time data. An *interpolated* value is
116 calculated from the data points on either side of the requested timestamp.

117 3.1.3

118 **EffectiveEndTime**

119 time immediately before *endTime*

120 Note 1 to entry: All *Aggregate* calculations include the *startTime* but exclude the *endTime*. However, it is
121 sometimes necessary to return an *Interpolated* End Bound as the value for an *Interval* with a timestamp that is in
122 the *interval*. *Servers* are expected to use the time immediately before *endTime* where the time resolution of the
123 *Server* determines the exact value (do not confuse this with hardware or operating system time resolution). For
124 example, if the *endTime* is 12:01:00, the time resolution is 1 second, then the *EffectiveEndTime* is 12:00:59. See
125 5.4.2.4.

126 If time is flowing backwards, *Servers* are expected to use the time immediately after *endTime* where the time
127 resolution of the *Server* determines the exact value.

128 **3.1.4**
 129 **extrapolated**
 130 data constructed from a discrete data set but is outside of the discrete data set

131 Note 1 to entry: It is similar to the process of interpolation, which constructs new points between known points,
 132 but its result is subject to greater uncertainty. *Extrapolated* data is used in cases where the requested time period
 133 falls farther into the future than the data available in the underlying system. See example in Table 1.

134 **3.1.5**
 135 **SlopedInterpolation**
 136 simple linear interpolation

137 Note 1 to entry: Compare to curve fitting using linear polynomials. See example in Table 1.

138 **3.1.6**
 139 **SteppedInterpolation**
 140 holding the last data point constant or interpolating the value based on a horizontal line fit

141 Note 1 to entry: Consider the following Table 1 of raw and *Interpolated/Extrapolated* values:

142 **Table 1 – Interpolation examples**

Timestamp	Raw Value	Sloped Interpolation	Stepped Interpolation
12:00:00	10		
12:00:05		15	10
12:00:08		18	10
12:00:10	20		
12:00:15		25	20
12:00:20	30		
		SlopedExtrapolation	SteppedExtrapolation
12:00:25		35	30
12:00:27		37	30

143

144 **3.1.7**
 145 **bounding values**
 146 values at the *startTime* and *endTime* needed for *Aggregates* to compute the result

147 Note 1 to entry: If *Raw data* does not exist at the *startTime* and *endTime* a value shall be estimated. There are
 148 two ways to determine *Bounding Values* for an interval. One way (called *Interpolated Bounding Values*) uses the
 149 first non-Bad data points found before and after the timestamp to estimate the bound. The other (called *Simple*
 150 *Bounding Values*) uses the data points immediately before and after the boundary timestamps to estimate the
 151 bound even if these points are Bad. Subclauses 3.1.8 and 3.1.9 describe the two different approaches in more
 152 detail.

153 In all cases the *TreatUncertainAsBad* (see 4.2.1.2) flag is used to determine whether Uncertain values are Bad or
 154 non-Bad.

155 If a Raw value was not found and a non-Bad bounding value exists the *Aggregate* Bits (see 5.3.3) are set to
 156 'Interpolated'.

157 When calculating *bounding values*, the value portion of *Raw data* that has Bad status is set to null. This means the
 158 value portion is not used in any calculation and a null is returned if the raw value is returned. The status portion is
 159 determined by the rules specified by the bound or *Aggregate*.

160 The *Interpolated Bounding Values* approach (see 3.1.8) is the same as what is used in Classic OPC Historical Data
 161 Access (HDA) and is important for applications such as advanced process control where having useful values at all
 162 times is important. The *Simple Bounding Values* approach (see 3.1.9) is new in this standard and is important for
 163 applications which shall produce regulatory reports and cannot use estimated values in place of Bad data.

164 **3.1.8**
 165 **interpolated bounding values**
 166 *bounding values* determined by a calculation using the nearest Good value

167 Note 1 to entry: *Interpolated Bounding Values* using *SlopedInterpolation* are calculated as follows:

- 168 • if a non-Bad Raw value exists at the timestamp then it is the bounding value;
- 169 • find the first non-Bad Raw value before the timestamp;
- 170 • find the first non-Bad Raw value after the timestamp;
- 171 • draw a line between before value and after value;
- 172 • use point where the line crosses the timestamp as an estimate of the bounding value.

173 The calculation can be expressed with the following formula:

$$174 \quad V_{\text{bound}} = (T_{\text{bound}} - T_{\text{before}}) \times (V_{\text{after}} - V_{\text{before}}) / (T_{\text{after}} - T_{\text{before}}) + V_{\text{before}}$$

175 where V_x is a value at 'x' and T_x is the timestamp associated with V_x .

176 If no non-Bad values exist before the timestamp the *StatusCode* is *Bad_NoData*. The *StatusCode* is
 177 *Uncertain_DataSubNormal* if any Bad values exist between the before value and after value. If either the before
 178 value or the after value are Uncertain the *StatusCode* is *Uncertain_DataSubNormal*. If the after value does not
 179 exist the before value shall be extrapolated using *SlopedExtrapolation* or *SteppedExtrapolation*.

180 The period of time that is searched to discover the Good values before and after the timestamp is *Server*
 181 dependent, but if a Good value is not found within some reasonable time range then the *Server* will assume it does
 182 not exist. The *Server* as a minimum should search a time range which is at least the size of the *ProcessingInterval*.

183 *Interpolated Bounding Values* using *SlopedExtrapolation* are calculated as follows:

- 184 • find the first non-Bad Raw value before timestamp;
- 185 • find the second non-Bad Raw value before timestamp;
- 186 • draw a line between these two values;
- 187 • extend the line to where it crosses the timestamp;
- 188 • use the point where the line crosses the timestamp as an estimate of the bounding value.

189 The formula is the same as the one used for *SlopedInterpolation*.

190 The *StatusCode* is always *Uncertain_DataSubNormal*. If only one non-Bad raw value can be found before the
 191 timestamp then *SteppedExtrapolation* is used to estimate the bounding value.

192 *Interpolated Bounding Values* using *SteppedInterpolation* are calculated as follows:

- 193 • if a non-Bad Raw value exists at the timestamp then it is the bounding value;
- 194 • find the first non-Bad Raw value before timestamp;
- 195 • use the value as an estimate of the bounding value.

196 The *StatusCode* is *Uncertain_DataSubNormal* if any Bad values exist between the before value and the timestamp.
 197 If no non-Bad Raw data exists before the timestamp then the *StatusCode* is *Bad_NoData*. If the value before the
 198 timestamp is Uncertain the *StatusCode* is *Uncertain_DataSubNormal*. The value after the timestamp is not needed
 199 when using *SteppedInterpolation*; however, if the timestamp is after the end of the data then the bounding value is
 200 treated as extrapolated and the *StatusCode* is *Uncertain_DataSubNormal*.

201 *SteppedExtrapolation* is a term that describes *SteppedInterpolation* when a timestamp is after the last value in the
 202 history collection.

203 3.1.9

204 simple bounding values

205 *bounding values* determined by a calculation using the nearest value

206 Note 1 to entry: *Simple Bounding Values* using *SlopedInterpolation* are calculated as follows:

- 207 • if any Raw value exists at the timestamp then it is the bounding value;
- 208 • find the first Raw value before timestamp;
- 209 • find the first Raw value after timestamp;
- 210 • if the value after the timestamp is Bad then the before value is the bounding value;
- 211 • draw a line between before value and after value;
- 212 • use point where the line crosses the timestamp as an estimate of the bounding value.

213 The formula is the same as the one used for *SlopedInterpolation* in Clause 3.1.5.

214 If a Raw value at the timestamp is Bad the *StatusCode* is *Bad_NoData*. If the value before the timestamp is Bad
 215 the *StatusCode* is *Bad_NoData*. If the value before the timestamp is Uncertain the *StatusCode* is
 216 *Uncertain_DataSubNormal*. If the value after the timestamp is Bad or Uncertain the *StatusCode* is
 217 *Uncertain_DataSubNormal*.

218 *Simple Bounding Values* using *SteppedInterpolation* are calculated as follows:

- 219 • if any Raw value exists at the timestamp then it is the bounding value;
- 220 • find the first Raw value before timestamp;
- 221 • if the value before timestamp is non-Bad then it is the bounding value.

222 If a Raw value at the timestamp is Bad the *StatusCode* is *Bad_NoData*. If the value before the timestamp is Bad
 223 the *StatusCode* is *Bad_NoData*. If the value before the timestamp is Uncertain the *StatusCode* is
 224 *Uncertain_DataSubNormal*.

225 If either bounding time of an interval is beyond the last data point then the *Server* may use extrapolation or return
 226 an error. If extrapolation is used by the server the type [*SteppedExtrapolation* or *SlopedExtrapolation*] of
 227 extrapolation is server specific.