

DIN EN 62541-11

ICS 35.200; 35.240.50

**OPC Unified Architecture –
Teil 11: Zugang zu historischen Daten
(IEC 62541-11:2015);
Englische Fassung EN 62541-11:2015**

OPC unified architecture –
Part 11: Historical Access
(IEC 62541-11:2015);
English version EN 62541-11:2015

Architecture unifiée OPC –
Partie 11: Accès à l’Historique
(IEC 62541-11:2015);
Version anglaise EN 62541-11:2015

Gesamtumfang 54 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE

Anwendungsbeginn

Anwendungsbeginn für die von CENELEC am 2015-04-29 angenommene Europäische Norm als DIN-Norm ist 2015-10-01.

Nationales Vorwort

Vorausgegangener Norm-Entwurf: E DIN EN 62541-11:2014-08.

Für dieses Dokument ist das nationale Arbeitsgremium K 931 „Systemaspekte“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE (www.dke.de) zuständig.

Die enthaltene IEC-Publikation wurde vom SC 65E „Devices and integration in enterprise systems“ erarbeitet.

Das IEC-Komitee hat entschieden, dass der Inhalt dieser Publikation bis zu dem Datum (stability date) unverändert bleiben soll, das auf der IEC-Website unter „<http://webstore.iec.ch>“ zu dieser Publikation angegeben ist. Zu diesem Zeitpunkt wird entsprechend der Entscheidung des Komitees die Publikation

- bestätigt,
- zurückgezogen,
- durch eine Folgeausgabe ersetzt oder
- geändert.

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf ein Dokument ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils aktuellste Ausgabe des in Bezug genommenen Dokuments.

Für den Fall einer datierten Verweisung im normativen Text bezieht sich die Verweisung immer auf die in Bezug genommene Ausgabe des Dokuments.

Der Zusammenhang der zitierten Dokumente mit den entsprechenden Deutschen Dokumenten ergibt sich, soweit ein Zusammenhang besteht, grundsätzlich über die Nummer der entsprechenden IEC-Publikation. Beispiel: IEC 60068 ist als EN 60068 als Europäische Norm durch CENELEC übernommen und als DIN EN 60068 ins Deutsche Normenwerk aufgenommen.

Da sich die Benutzer der vorliegenden Norm der englischen Sprache als Fachsprache bedienen, wird die Englische Fassung der EN 62541-11 veröffentlicht. Zu deren Abschnitt 1, der den Anwendungsbereich festlegt, und Abschnitt 3, der die Begriffe festlegt, wurde eine Übersetzung angefertigt und als Nationaler Anhang NA der vorliegenden Norm hinzugefügt. Für viele der verwendeten Begriffe existieren keine gebräuchlichen deutschen Benennungen, da sich die deutschen Anwender in der Regel ebenfalls der englischen Benennungen bedienen. Diese Norm steht nicht in unmittelbarem Zusammenhang mit Rechtsvorschriften und ist nicht als Sicherheitsnorm anzusehen.

Das Original-Dokument enthält Bilder in Farbe, die in der Papierversion in einer Graustufen-Darstellung wiedergegeben werden. Elektronische Versionen dieses Dokuments enthalten die Bilder in der originalen Farbdarstellung.

Nationaler Anhang NA (informativ)

1 Anwendungsbereich

Dieser Teil der Normenreihe IEC 62541 ist Teil der gesamten Normenreihe zur OPC Unified Architecture (vereinheitlichte Architektur der plattformunabhängigen Kommunikation) (OPC-UA)) und legt das Informationsmodell fest, welches im Zusammenhang mit dem Zugriff auf historische Daten (en: Historical Access (HA)) steht. Es umfasst insbesondere zusätzliche und ergänzende Festlegungen der *KnotenKlassen* (en: *NodeClasses*) und *Attribute* (en: *Attributes*), welche für den Zugriff auf historische Daten erforderlich sind, zusätzliche *Eigenschaften* (en: *Properties*) und weitere Informationen und Verhalten.

Das vollständige Adressraummodell, einschließlich aller *KnotenKlassen* und *Attribute* ist in IEC 62541-3 festgelegt. Das vorgegebene Informationsmodell ist in IEC 62541-5 festgelegt. Die *Dienste* für das Erkennen von und den Zugriff auf historische Daten und Ereignisse und Festlegung der *ErweiterbarerParameterTypen* (en: *ExtensibleParameterTypes*), sind in IEC 62541-4.

Diese Norm schließt die Funktionalität für die Berechnung und Rückführung von Aggregaten (en: *Aggregates*) ein, wie z. B. Minimum, Maximum, Mittelwert, usw. Das *Informationsmodell* und die genaue Funktionsweise von *Aggregaten* sind in IEC 62541-13 festgelegt.

3 Begriffe und Abkürzungen

3.1 Begriffe

Für die Anwendung dieses Dokuments gelten die Begriffe nach IEC/TR 62541-1, IEC 62541-3, IEC 62541-4 und IEC 62541-13 und die folgenden Begriffe.

3.1.1

Annotation

(en: Annotation)

Metadaten, die mit einem Element zu einem bestimmten Zeitpunkt im Zusammenhang stehen

ANMERKUNG 1 zum Begriff: Eine *Annotation* entspricht Metadaten, die mit einem Element zu einem bestimmten Zeitpunkt im Zusammenhang stehen. Dabei muss es sich nicht um einen zu diesem Zeitpunkt gespeicherten Wert handeln.

3.1.2

GrenzWerte

(en: BoundingValues)

Werte, die mit der Start- und Endzeit verbunden sind

ANMERKUNG 1 zum Begriff: *GrenzWerte* sind die Werte, die der Start- und Endzeit eines *VerarbeitungsIntervalls* zugeordnet sind, welche für das Lesen aus dem Historian angegeben werden. *GrenzWerte* dürfen von *Clients* im Zuge einer Abfrage von Rohdaten für einen bestimmten Zeitraum gefordert werden, um die Start- und Endwerte zu bestimmen. Wenn am Start- oder Endpunkt ein Rohdatenwert vorhanden ist, wird dieser als Grenzwert betrachtet, auch wenn er Teil der Datenabfrage ist. Wenn am Start- oder Endpunkt kein Rohdatenwert vorhanden ist, wird der *GrenzWert* durch den *Server* festgelegt, für den unter Umständen Daten von einem Datenpunkt außerhalb des abgefragten Bereiches erforderlich sind. Siehe 4.4 für Einzelheiten zur Verwendung von *GrenzWerten*.

3.1.3

HistorischerKnoten

(en: HistoricalNode)

Objekt, *Variable*, *Eigenschaft* oder *Sicht* (en: *View*) in dem *AdressRaum*, für den ein *Client* Zugriff auf historische Daten oder Ereignisse hat

ANMERKUNG 1 zum Begriff: Die Bezeichnung *HistorischerKnoten* wird in diesem Dokument verwendet, um ein *Objekt*, eine *Variable*, eine *Eigenschaft* oder eine *Sicht* in dem *AdressRaum* darzustellen, für den ein *Client* historische Daten oder Ereignisse lesen und/oder aktualisieren kann. Die Begriffe „Historie des *HistorischenKnotens*“ oder „Historie eines *HistorischenKnotens*“ beziehen sich auf die Zeitreihendaten oder *Ereignisse*, die für diesen *HistorischenKnoten* gespeichert wurden. Der Begriff *HistorischerKnoten* bezieht sich sowohl auf *HistorischeDatenKnoten* als auch auf *HistorischesEreignisKnoten*.

3.1.4

HistorischerDatenKnoten

(en: HistoricalDataNode)

Variable oder *Eigenschaft* in dem *AdressRaum*, für den ein *Client* Zugang zu historischen Daten hat

ANMERKUNG 1 zum Begriff: Ein *HistorischerDatenKnoten* stellt eine beliebige *Variable* oder *Eigenschaft* in dem *AdressRaum* dar, für den ein *Client* historische Daten lesen und/oder aktualisieren kann. Die Ausdrücke „Historie des *HistorischenDatenKnotens*“ oder „Historie eines *HistorischenDatenKnotens*“ beziehen sich auf die Zeitreihendaten, die für diesen *HistorischenKnoten* gespeichert wurden. Beispiele für derartige Daten sind:

- Gerätedaten (wie Temperatursensoren);
- berechnete Daten;
- Statusinformationen (offen/geschlossen, in Bewegung);
- sich dynamisch ändernde Systemdaten (wie Aktienkurse);
- Diagnosedaten.

Der Begriff *HistorischerDatenKnoten* wird zur Referenzierung von Aspekten der Spezifikation verwendet, die sich nur auf den Zugang zu historische Daten beziehen.

3.1.5

HistorischerEreignisKnoten

(en: HistoricalEventNode)

Objekt oder *Sicht* in dem *AdressRaum*, für den ein *Client* Zugriff auf historische Ereignisse hat

ANMERKUNG 1 zum Begriff: Die Ausdrücke „Historie des *HistorischenEreignisKnotens*“ oder „Historie eines *HistorischenEreignisKnotens*“ beziehen sich auf die Zeitreihen-Ereignisse, die in einem Historie-System gespeichert wurden. Beispiele für derartige Daten sind:

- Benachrichtigungen;
- System-Alarme;
- *Ereignisse* von Operatoraktionen;
- System-Auslöser (wie zu verarbeitende neue Aufträge).

Der Begriff *HistorischerEreignisKnoten* wird zur Referenzierung von Aspekten der Spezifikation verwendet, die sich nur auf den Zugriff auf historische *Ereignisse* beziehen.

3.1.6

modifizierte Werte

(en: Modified values)

Wert eines *HistorischenDatenKnotens*, der nach der Speicherung im Historian geändert wurde (oder manuell eingefügt oder gelöscht wurde)

ANMERKUNG 1 zum Begriff: Bei einigen *Servern* wird der Wert eines Labordateneintrags nicht als *modifizierter Wert* angesehen; wenn allerdings ein Benutzer einen Laborwert korrigiert, würde der ursprüngliche Wert als *modifizierter Wert* betrachtet und bei einer Abfrage von *modifizierten Werten* zurückgegeben werden. Auch ein manuell eingefügter Wert, der von einem Standardsammlungssystem übersehen wurde, darf als *modifizierter Wert* betrachtet werden. Sofern nichts anderes angegeben ist, werden alle historischen *Dienste* für den aktuellen, oder neuesten, Wert des angegebenen *HistorischenDatenKnotens* zu dem angegebenen Zeitstempel ausgeführt. Abfragen von *modifizierten Werten* werden eingesetzt, um auf ersetzte, gelöschte oder eingefügte Werte zugreifen zu können. Es obliegt einem System, zu definieren, wann ein Wert als *modifizierter Wert* angesehen wird. Jedes Mal, wenn ein *Server* Daten modifiziert hat, die

für einen Eintrag in der historischen Datensammlung verfügbar sind, muss im *StatusCode* das *ZusatzDaten*-Bit gesetzt werden.

3.1.7

Rohdaten

(en: Raw Data)

Daten, die innerhalb des Historians für einen *HistorischenDatenKnoten* gespeichert werden

ANMERKUNG 1 zum Begriff: Die Daten können allen für den *DatenWert* erfassten Daten oder einer Teilmenge dieser Daten entsprechen; dies hängt von dem Historian und den Speicherregeln ab, die bei der Speicherung der Werte des Elementes aufgerufen wurden.

3.1.8

StartZeit/EndZeit

(en: StartTime/EndTime)

die Grenzen einer Historienabfrage, welche die Zeitdomäne festlegen

ANMERKUNG 1 zum Begriff: Bei allen Abfragen wird ein Wert, der auf die Endzeit der Zeitdomäne fällt, nicht in die Domäne eingeschlossen, damit jeder Wert in der historischen Datensammlung in Abfragen für aufeinanderfolgende, fortlaufende Zeitdomänen genau einmal enthalten ist.

3.1.9

ZeitDomäne

(en: TimeDomain)

Zeitintervall, das durch eine bestimmte Abfrage oder Antwort abgedeckt ist

ANMERKUNG 1 zum Begriff: Im Allgemeinen gilt: wenn die *StartZeit* vor der *EndZeit* liegt oder mit dieser zusammenfällt, wird angenommen, dass die *ZeitDomäne* bei der *StartZeit* beginnt und unmittelbar vor der *EndZeit* endet; wenn die *EndZeit* vor der *StartZeit* liegt, wird angenommen, dass die *ZeitDomäne* dennoch bei der *StartZeit* beginnt und unmittelbar vor der *EndZeit* endet, wobei die Zeit für diese spezielle Abfrage und Antwort „rückwärts läuft“. In beiden Fällen wird jeder Wert, der direkt mit der *EndZeit* der *ZeitDomäne* zusammenfällt, nicht in die *ZeitDomäne* eingeschlossen. Siehe Beispiele in 4.4. *GrenzWerte* haben die in 4.4 beschriebene Auswirkung auf die *ZeitDomäne*.

Alle Zeitstempel, die mit dem *DatenTyp UtcZeit* (en: UtcTime) dargestellt werden können, sind gültige Zeitstempel, für die der *Server* keinen ungültigen Argumentergebniscode zurückgeben darf, wenn der Zeitstempel außerhalb des Bereiches liegt, zu welchem dem *Server* Daten vorliegen. Siehe IEC 62541-3 für eine Beschreibung des Bereiches und der Granularität dieses *DatenTyps*. Von *Servern* wird erwartet, dass sie kontrolliert auf Zeitstempel außerhalb der *GrenzWerte* reagieren und die richtigen *StatusCodes* an den *Client* zurückgeben.

3.1.10

strukturierte Historiedaten

(en: Structured History Data)

strukturierte Daten, die in einer historischen Datensammlung gespeichert sind, wobei Teile der Struktur zur eindeutigen Kennzeichnung der Daten innerhalb der Datensammlung verwendet werden

ANMERKUNG 1 zum Begriff: Die meisten historischen Datenanwendungen gehen von einem einzigen aktuellen Wert je Zeitstempel aus. Aus diesem Grund wird der Zeitstempel der Daten als eindeutige Kennung für diesen Wert betrachtet. Einige Daten oder Metadaten, wie *Annotationen*, können zulassen, dass mehrere Werte zu einem einzigen Zeitstempel vorhanden sind. In derartigen Fällen würde der *Server* einen oder mehrere Parameter des Eintrags der strukturierten Historiedaten verwenden, um jedes Element innerhalb der historischen Datensammlung eindeutig zu kennzeichnen. *Annotationen* sind Beispiele für *strukturierte Historiedaten*.

– Leerseite –

English Version

OPC unified architecture - Part 11: Historical Access
(IEC 62541-11:2015)

Architecture unifiée OPC - Partie 11: Accès à l'Histoire
(IEC 62541-11:2015)

OPC Unified Architecture - Teil 11: Zugang zu historischen
Daten
(IEC 62541-11:2015)

This European Standard was approved by CENELEC on 2015-04-29. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.



European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

CEN-CENELEC Management Centre: Avenue Marnix 17, B-1000 Brussels

Foreword

The text of document 65E/380/CDV, future edition 1 of IEC 62541-11, prepared by SC 65E "Devices and integration in enterprise systems", of IEC/TC 65 "Industrial-process measurement, control and automation" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN 62541-11:2015.

The following dates are fixed:

- latest date by which the document has to be implemented at national level by publication of an identical national standard or by endorsement (dop) 2016-01-29
- latest date by which the national standards conflicting with the document have to be withdrawn (dow) 2018-04-29

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CENELEC [and/or CEN] shall not be held responsible for identifying any or all such patent rights.

This document has been prepared under a mandate given to CENELEC by the European Commission and the European Free Trade Association, and supports essential requirements of EU Directive(s).

Endorsement notice

The text of the International Standard IEC 62541-11:2015 was approved by CENELEC as a European Standard without any modification.

In the official version, for Bibliography, the following notes have to be added for the standards indicated:

IEC/TR 62541-2	NOTE	Harmonized as CLC/TR 62541-2.
IEC 62541-6	NOTE	Harmonized as EN 62541-6.
IEC 62541-7	NOTE	Harmonized as EN 62541-7.
IEC 62541-9	NOTE	Harmonized as EN 62541-9.

CONTENTS

1	Scope.....	6
2	Normative references	6
3	Terms, definitions, and abbreviations	6
3.1	Terms and definitions	6
3.2	Abbreviations	8
4	Concepts.....	8
4.1	General.....	8
4.2	Data architecture.....	8
4.3	Timestamps	9
4.4	Bounding Values and time domain	10
4.5	Changes in AddressSpace over time	12
5	Historical Information Model.....	12
5.1	HistoricalNodes.....	12
5.1.1	General.....	12
5.1.2	Annotations Property.....	12
5.2	HistoricalDataNodes.....	13
5.2.1	General.....	13
5.2.2	HistoricalDataConfigurationType.....	13
5.2.3	HasHistoricalConfiguration ReferenceType	14
5.2.4	Historical Data Configuration Object	15
5.2.5	HistoricalDataNodes Address Space Model	16
5.2.6	Attributes	16
5.3	HistoricalEventNodes	17
5.3.1	General.....	17
5.3.2	HistoricalEventFilter Property	17
5.3.3	HistoricalEventNodes Address Space Model	17
5.3.4	HistoricalEventNodes Attributes.....	18
5.4	Exposing supported functions and capabilities	18
5.4.1	General.....	18
5.4.2	HistoryServerCapabilitiesType.....	19
5.5	Annotation DataType.....	21
5.6	Historical Audit Events	22
5.6.1	General.....	22
5.6.2	AuditHistoryEventUpdateEventType	22
5.6.3	AuditHistoryValueUpdateEventType	23
5.6.4	AuditHistoryDeleteEventType	24
5.6.5	AuditHistoryRawModifyDeleteEventType.....	24
5.6.6	AuditHistoryAtTimeDeleteEventType.....	25
5.6.7	AuditHistoryEventDeleteEventType.....	25
6	Historical Access specific usage of Services	26
6.1	General.....	26
6.2	Historical Nodes StatusCodes	26
6.2.1	Overview.....	26
6.2.2	Operation level result codes	26

6.2.3	Semantics changed	28
6.3	Continuation Points	28
6.4	HistoryReadDetails parameters	29
6.4.1	Overview	29
6.4.2	ReadEventDetails structure	29
6.4.3	ReadRawModifiedDetails structure	31
6.4.4	ReadProcessedDetails structure	33
6.4.5	ReadAtTimeDetails structure	34
6.5	HistoryData parameters returned	35
6.5.1	Overview	35
6.5.2	HistoryData type	35
6.5.3	HistoryModifiedData type	35
6.5.4	HistoryEvent type	36
6.6	HistoryUpdateType Enumeration	36
6.7	PerformUpdateType Enumeration	36
6.8	HistoryUpdateDetails parameter	37
6.8.1	Overview	37
6.8.2	UpdateDataDetails structure	38
6.8.3	UpdateStructureDataDetails structure	39
6.8.4	UpdateEventDetails structure	40
6.8.5	DeleteRawModifiedDetails structure	42
6.8.6	DeleteAtTimeDetails structure	43
6.8.7	DeleteEventDetails structure	44
Annex A (informative)	Client conventions	45
A.1	How clients may request timestamps	45
A.2	Determining the first historical data point	46
Bibliography		47
Annex ZA (normative)	Normative references to international publications with their corresponding European publications	48
Figure 1	Possible OPC UA Server supporting Historical Access	9
Figure 2	ReferenceType hierarchy	15
Figure 3	Historical Variable with Historical Data Configuration and Annotations	16
Figure 4	Representation of an Event with History in the AddressSpace	18
Figure 5	Server and HistoryServer Capabilities	19
Table 1	Bounding Value examples	11
Table 2	Annotations Property	12
Table 3	HistoricalDataConfigurationType definition	13
Table 4	ExceptionDeviationFormat Values	14
Table 5	HasHistoricalConfiguration ReferenceType	15
Table 6	Historical Access configuration definition	15
Table 7	Historical Events Properties	17
Table 8	HistoryServerCapabilitiesType Ddefinition	20
Table 9	Annotation Structure	22
Table 10	AuditHistoryEventUpdateEventType definition	22

Table 11 – AuditHistoryValueUpdateEventType definition	23
Table 12 – AuditHistoryDeleteEventType definition	24
Table 13 – AuditHistoryRawModifyDeleteEventType definition	24
Table 14 – AuditHistoryAtTimeDeleteEventType definition	25
Table 15 – AuditHistoryEventDeleteEventType definition	25
Table 16 – Bad operation level result codes	27
Table 17 – Good operation level result codes	27
Table 18 – HistoryReadDetails parameterTypelds	29
Table 19 – ReadEventDetails	30
Table 20 – ReadRawModifiedDetails	31
Table 21 – ReadProcessedDetails.....	33
Table 22 – ReadAtTimeDetails	35
Table 23 – HistoryData Details	35
Table 24 – HistoryModifiedData Details	36
Table 25 – HistoryEvent Details	36
Table 26 – HistoryUpdateType Enumeration	36
Table 27 – PerformUpdateType Enumeration	36
Table 28 – HistoryUpdateDetails parameter Typelds.....	37
Table 29 – UpdateDataDetails.....	38
Table 30 – UpdateStructureDataDetails.....	39
Table 31 – UpdateEventDetails	41
Table 32 – DeleteRawModifiedDetails	43
Table 33 – DeleteAtTimeDetails	43
Table 34 – DeleteEventDetails	44
Table A.1 – Time keyword definitions	46
Table A.2 –Time offset definitions	46

OPC UNIFIED ARCHITECTURE –

Part 11: Historical Access

1 Scope

This part of IEC 62541 is part of the overall OPC Unified Architecture standard series and defines the *information model* associated with Historical Access (HA). It particularly includes additional and complementary descriptions of the *NodeClasses* and *Attributes* needed for Historical Access, additional standard *Properties*, and other information and behaviour.

The complete *AddressSpace* Model including all *NodeClasses* and *Attributes* is specified in IEC 62541-3. The predefined *Information Model* is defined in IEC 62541-5. The *Services* to detect and access historical data and events, and description of the *ExtensibleParameter* types are specified in IEC 62541-4.

This standard includes functionality to compute and return *Aggregates* like minimum, maximum, average etc. The *Information Model* and the concrete working of *Aggregates* are defined in IEC 62541-13.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

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

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

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

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

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

IEC 62541-13, *OPC Unified Architecture – Part 13: Aggregates*

3 Terms, definitions, and abbreviations

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC TR 62541-1, IEC 62541-3, IEC 62541-4, and IEC 62541-13 as well as the following apply.

3.1.1

Annotation

metadata associated with an item at a given instance in time

Note 1 to entry: An *Annotation* is metadata that is associated with an item at a given instance in time. There does not have to be a value stored at that time.

3.1.2

BoundingValues

values associated with the starting and ending time

Note 1 to entry: *BoundingValues* are the values that are associated with the starting and ending time of a *ProcessingInterval* specified when reading from the historian. *BoundingValues* may be required by *Clients* to determine the starting and ending values when requesting *raw data* over a time range. If a *raw data* value exists at the start or end point, it is considered the bounding value even though it is part of the data request. If no *raw data* value exists at the start or end point, then the *Server* will determine the boundary value, which may require data from a data point outside of the requested range. See 4.4 for details on using *BoundingValues*.

3.1.3

HistoricalNode

Object, Variable, Property or View in the *AddressSpace* where a *Client* can access historical data or *Events*

Note 1 to entry: A *HistoricalNode* is a term used in this document to represent any *Object, Variable, Property or View* in the *AddressSpace* for which a *Client* may read and/or update historical data or *Events*. The terms "*HistoricalNode's history*" or "history of a *HistoricalNode*" will refer to the time series data or *Events* stored for this *HistoricalNode*. The term *HistoricalNode* refers to both *HistoricalDataNodes* and *HistoricalEventNodes*.

3.1.4

HistoricalDataNode

Variable or Property in the *AddressSpace* where a *Client* can access historical data

Note 1 to entry: A *HistoricalDataNode* represents any *Variable or Property* in the *AddressSpace* for which a *Client* may read and/or update historical data. "*HistoricalDataNode's history*" or "history of a *HistoricalDataNode*" refers to the time series data stored for this *HistoricalNode*. Examples of such data are:

- device data (like temperature sensors),
- calculated data,
- status information (open/closed, moving),
- dynamically changing system data (like stock quotes),
- diagnostic data.

The term *HistoricalDataNodes* is used when referencing aspects of the standard that apply to accessing historical data only.

3.1.5

HistoricalEventNode

Object or View in the *AddressSpace* for which a *Client* can access historical *Events*

Note 1 to entry: "*HistoricalEventNode's history*" or "history of a *HistoricalEventNode*" refers to the time series *Events* stored in some historical system. Examples of such data are:

- *Notifications*,
- system *Alarms*,
- operator action *Events*,
- system triggers (such as new orders to be processed).

The term *HistoricalEventNode* is used when referencing aspects of the standard that apply to accessing historical *Events* only.

3.1.6

modified values

HistoricalDataNode's value that has been changed (or manually inserted or deleted) after it was stored in the historian

Note 1 to entry: For some *Servers*, a lab data entry value is not a *modified value*, but if a user corrects a lab value, the original value would be considered a *modified value*, and would be returned during a request for *modified values*. Also manually inserting a value that was missed by a standard collection system may be considered a *modified value*. Unless specified otherwise, all historical *Services* operate on the current, or most recent, value for the specified *HistoricalDataNode* at the specified timestamp. Requests for *modified values* are used to access values that have been

superseded, deleted or inserted. It is up to a system to determine what is considered a *modified value*. Whenever a *Server* has modified data available for an entry in the historical collection it shall set the *ExtraData* bit in the *StatusCode*.

3.1.7

raw data

data that is stored within the historian for a *HistoricalDataNode*

Note 1 to entry: The data may be all data collected for the *DataValue* or it may be some subset of the data depending on the historian and the storage rules invoked when the item's values were saved.

3.1.8

StartTime/EndTime

bounds of a history request which define the time domain

Note 1 to entry: For all requests, a value falling at the end time of the time domain is not included in the domain, so that requests made for successive, contiguous time domains will include every value in the historical collection exactly once.

3.1.9

TimeDomain

interval of time covered by a particular request, or response

Note 1 to entry: In general, if the start time is earlier than or the same as the end time, the time domain is considered to begin at the start time and end just before the end time; if the end time is earlier than the start time, the time domain still begins at the start time and ends just before the end time, with time "running backward" for the particular request and response. In both cases, any value which falls exactly at the end time of the *TimeDomain* is not included in the *TimeDomain*. See the examples in 4.4. *BoundingValues* effect the time domain as described in 4.4.

All timestamps which can legally be represented in a *UtcTime DataType* are valid timestamps, and the *Server* may not return an invalid argument result code due to the timestamp being outside of the range for which the *Server* has data. See IEC 62541-3 for a description of the range and granularity of this *DataType*. *Servers* are expected to handle out-of-bounds timestamps gracefully, and return the proper *StatusCodes* to the *Client*.

3.1.10

Structured History Data

structured data stored in a history collection where parts of the structure are used to uniquely identify the data within the data collection

Note 1 to entry: Most historical data applications assume only one current value per timestamp. Therefore the timestamp of the data is considered the unique identifier for that value. Some data or meta data such as *Annotations* may permit multiple values to exist at a single timestamp. In such cases the *Server* would use one or more parameters of the *Structured History Data* entry to uniquely identify each element within the history collection. *Annotations* are examples of *Structured History Data*.

3.2 Abbreviations

DA	Data Access
HA	Historical Access
HDA	Historical Data Access
UA	Unified Architecture

4 Concepts

4.1 General

This standard defines the handling of historical time series data and historical *Event* data in the OPC Unified Architecture. Included is the specification of the representation of historical data and *Events* in the *AddressSpace*.

4.2 Data architecture

A *Server* supporting Historical Access provides *Clients* with transparent access to different historical data and/or historical *Event* sources (e.g. process historians, event historians, etc.).

The historical data or *Events* may be located in a proprietary data collection, database or a short term buffer within the memory. A *Server* supporting Historical Access will provide historical data and *Events* for all or a subset of the available *Variables*, *Objects*, *Properties* or *Views* within the *Server AddressSpace*.

Figure 1 illustrates how the *AddressSpace* of a UA *Server* might consist of a broad range of different historical data and/or historical *Event* sources.

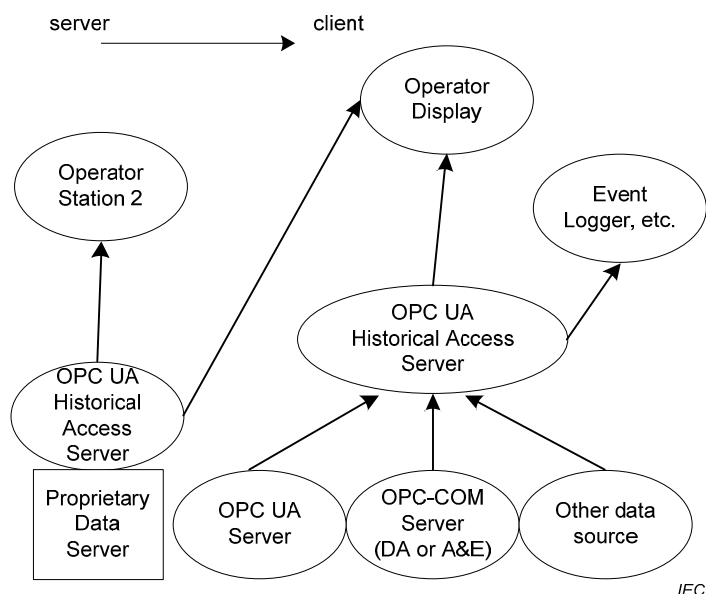


Figure 1 – Possible OPC UA Server supporting Historical Access

The *Server* may be implemented as a standalone OPC UA *Server* that collects data from another OPC UA *Server* or another data source. The *Client* that references the OPC UA *Server* supporting Historical Access for historical data may be simple trending packages that just desire values over a given time frame or they may be complex reports that require data in multiple formats.

4.3 Timestamps

The nature of OPC UA Historical Access requires that a single timestamp reference be used to relate the multiple data points, and the *Client* may request which timestamp will be used as the reference. See IEC 62541-4 for details on the *TimestampsToReturn* enumeration. An OPC UA *Server* supporting Historical Access will treat the various timestamp settings as described below. A *HistoryRead* with invalid settings will be rejected with *Bad_TimestampsToReturnInvalid* (see IEC 62541-4).

For *HistoricalDataNodes*, the *SourceTimestamp* is used to determine which historical data values are to be returned.

The request is in terms of *SourceTimestamp* but the reply could be in *SourceTimestamp*, *ServerTimestamp* or both timestamps. If the reply has the *Server* timestamp the timestamps could fall outside of the range of the requested time.

SOURCE_0	Return the <i>SourceTimestamp</i> .
SERVER_1	Return the <i>ServerTimestamp</i> .
BOTH_2	Return both the <i>SourceTimestamp</i> and <i>ServerTimestamp</i> .
NEITHER_3	This is not a valid setting for any <i>HistoryRead</i> accessing <i>HistoricalDataNodes</i> .

Any reference to timestamps in this context throughout this standard will represent either *ServerTimestamp* or *SourceTimestamp* as dictated by the type requested in the *HistoryRead Service*. Some *Servers* may not support historizing both *SourceTimestamp* and *ServerTimestamp*, but it is expected that all *Servers* will support historizing *SourceTimestamp* (see IEC 62541-7 for details on *Server Profiles*).

If a request is made requesting both *ServerTimestamp* and *SourceTimestamp* and the *Server* is only collecting the *SourceTimestamp* the *Server* shall return *Bad_TimestampsToReturnInvalid*.

For *HistoricalEventNodes* this parameter does not apply. This parameter is ignored since the entries returned are dictated by the *Event Filter*. See IEC 62541-4 for details.

4.4 Bounding Values and time domain

When accessing *HistoricalDataNodes* via the *HistoryRead Service*, requests can set a flag, *returnBounds*, indicating that *BoundingValues* are requested. For a complete description of the *Extensible Parameter HistoryReadDetails* that include *StartTime*, *EndTime* and *NumValuesPerNode*, see 6.4. The concept of Bounding Values and how they affect the time domain that is requested as part of the *HistoryRead* request is further explained in 4.4. 4.4 also provides examples of *TimeDomains* to further illustrate the expected behaviour.

When making a request for historical data using the *HistoryRead Service*, the required parameters include at least 2 of these three parameters: *startTime*, *endTime* and *numValuesPerNode*. What is returned when Bounding Values are requested varies according to which of these parameters are provided. For a historian that has values stored at 5:00, 5:02, 5:03, 5:05 and 5:06, the data returned when using the *Read Raw* functionality is given by Table 1. In the table, FIRST stands for a tuple with a value of null, a timestamp of the specified *StartTime*, and a *StatusCode* of *Bad_BoundNotFound*. LAST stands for a tuple with a value of null, a timestamp of the specified *EndTime*, and a *StatusCode* of *Bad_BoundNotFound*.

In some cases, attempting to locate bounds, particularly FIRST or LAST points, may be resource intensive for *Servers*. Therefore how far back or forward to look in history for Bounding Values is *Server* dependent, and the *Server* search limits may be reached before a bounding value can be found. There are also cases, such as reading *Annotations* or *Attribute* data where Bounding Values may not be appropriate. For such use cases it is permissible for the *Server* to return a *StatusCode* of *Bad_BoundNotSupported*.

Table 1 – Bounding Value examples

Start Time	End Time	numValuesPerNode	Bounds	Data Returned
5:00	5:05	0	Yes	5:00, 5:02, 5:03, 5:05
5:00	5:05	0	No	5:00, 5:02, 5:03
5:01	5:04	0	Yes	5:00, 5:02, 5:03, 5:05
5:01	5:04	0	No	5:02, 5:03
5:05	5:00	0	Yes	5:05, 5:03, 5:02, 5:00
5:05	5:00	0	No	5:05, 5:03, 5:02
5:04	5:01	0	Yes	5:05, 5:03, 5:02, 5:00
5:04	5:01	0	No	5:03, 5:02
4:59	5:05	0	Yes	FIRST, 5:00, 5:02, 5:03, 5:05
4:59	5:05	0	No	5:00, 5:02, 5:03
5:01	5:07	0	Yes	5:00, 5:02, 5:03, 5:05, 5:06, LAST
5:01	5:07	0	No	5:02, 5:03, 5:05, 5:06
5:00	5:05	3	Yes	5:00, 5:02, 5:03
5:00	5:05	3	No	5:00, 5:02, 5:03
5:01	5:04	3	Yes	5:00, 5:02, 5:03
5:01	5:04	3	No	5:02, 5:03
5:05	5:00	3	Yes	5:05, 5:03, 5:02
5:05	5:00	3	No	5:05, 5:03, 5:02
5:04	5:01	3	Yes	5:05, 5:03, 5:02
5:04	5:01	3	No	5:03, 5:02
4:59	5:05	3	Yes	FIRST, 5:00, 5:02
4:59	5:05	3	No	5:00, 5:02, 5:03
5:01	5:07	3	Yes	5:00, 5:02, 5:03
5:01	5:07	3	No	5:02, 5:03, 5:05
5:00	UNSPECIFIED	3	Yes	5:00, 5:02, 5:03
5:00	UNSPECIFIED	3	No	5:00, 5:02, 5:03
5:00	UNSPECIFIED	6	Yes	5:00, 5:02, 5:03, 5:05, 5:06, LAST ^a
5:00	UNSPECIFIED	6	No	5:00, 5:02, 5:03, 5:05, 5:06
5:07	UNSPECIFIED	6	Yes	5:06, LAST
5:07	UNSPECIFIED	6	No	NODATA
UNSPECIFIED	5:06	3	Yes	5:06,5:05,5:03
UNSPECIFIED	5:06	3	No	5:06,5:05,5:03
UNSPECIFIED	5:06	6	Yes	5:06,5:05,5:03,5:02,5:00,FIRST
UNSPECIFIED	5:06	6	No	5:06, 5:05, 5:03, 5:02, 5:00
UNSPECIFIED	4:48	6	Yes	5:00, FIRST
UNSPECIFIED	4:48	6	No	NODATA
4:48	4:48	0	Yes	FIRST,5:00
4:48	4:48	0	No	NODATA
4:48	4:48	1	Yes	FIRST
4:48	4:48	1	No	NODATA
4:48	4:48	2	Yes	FIRST,5:00
5:00	5:00	0	Yes	5:00,5:02 ^c
5:00	5:00	0	No	5:00
5:00	5:00	1	Yes	5:00
5:00	5:00	1	No	5:00
5:01	5:01	0	Yes	5:00, 5:02