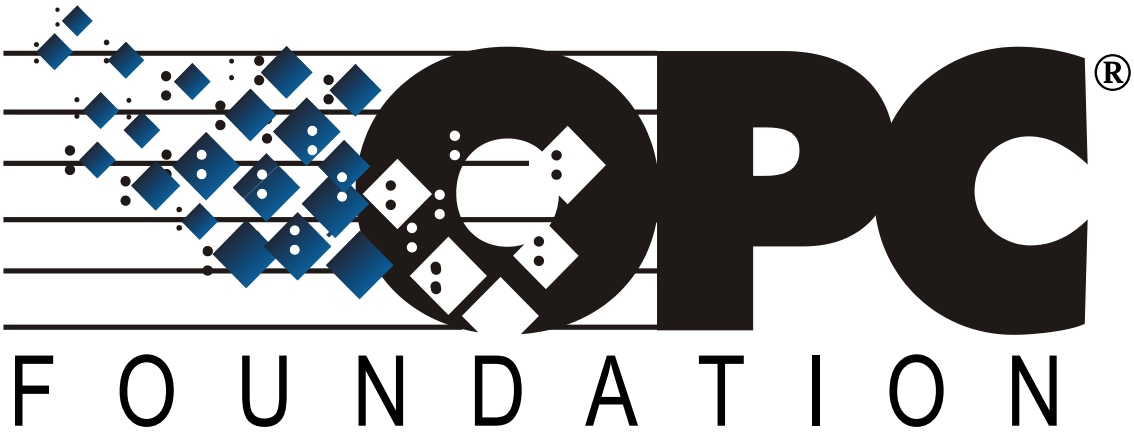


# OPC Unified Architecture

# Specification

# Errata

# Release 1.03.9

# 2022-06-14

| | | | |
|---|---|---|---|
| Specification Type: | Industry Standard Specification | Comments: | Report or view errata: http://www.opcfoundation.org/errata |
| Title: | OPC Unified Architecture<br><br>Errata : | Date: | 2022-06-14 |
| Version: | Release 1.03.9 | Software: | MS-Word |
| | | Source: | OPC UA 1.03.9 Specification Errata.docx |
| Author: | OPC Foundation | Status: | Release |

# CONTENTS

Page

## FIGURES

**No table of figures entries found.**

**TABLES**

# OPC FOUNDATION

# UNIFIED ARCHITECTURE –

## FOREWORD

This specification is the specification for developers of OPC UA applications. The specification is a result of an analysis and design process to develop a standard interface to facilitate the development of applications by multiple vendors that shall inter-operate seamlessly together.



## AGREEMENT OF USE

GENERAL PROVISIONS

Should any provision of this Agreement be held to be void, invalid, unenforceable or illegal by a court, the validity and enforceability of the other provisions shall not be affected thereby.

This Agreement shall be governed by and construed under the laws of the State of Minnesota, excluding its choice or law rules.

This Agreement embodies the entire understanding between the parties with respect to, and supersedes any prior understanding or agreement (oral or written) relating to, this specification.

ISSUE REPORTING

The OPC Foundation strives to maintain the highest quality standards for its published specifications, hence they undergo constant review and refinement. Readers are encouraged to report any issues and view any existing errata here: http://www.opcfoundation.org/errata

# OPC Unified Architecture Specification

# Errata

## Scope

This Errata document contains all of the known corrections to OPC UA Specification Parts 1 through 13 for version 1.03. This document is updated regularly when issues are found between major releases of the Specification.

## OPC UA Specification: Part 3 – Information Model

<table>
<tr><td>Topic</td><td>Correct inconsistence between model compilers and the specification.</td></tr>
<tr><td>Errata Version</td><td>1.03.1</td></tr>
<tr><td>Spec Reference</td><td>Part 3<br>C.3 – Standard Type Descriptions</td></tr>
<tr><td>Mantis Reference</td><td>0003750</td></tr>
<tr><td>Problem Statement</td><td>The model compiler implementation of String and WideString standard types is inconsistent with the definitions of these types in Part 3 Table C.3.</td></tr>
<tr><td>Solution</td><td>The model compilers definition of String and WideString will continue to be used. Many existing applications are built based on the model compiler’s implementation and therefore will not be impacted.<br><br>Note: the standard type descriptions have moved from Part 3 Table C.3 to Part 5 Table E.9 in version 1.04.<br><br>The standard types CharArray, WideChar, and WideString have been removed from the specification.<br><br>The description of the standard type String has been changed to the version 1.03 description of CharArray with a clarification. The clarification defines the size of the String in terms of UTF-8 Code Units (bytes) rather than the use of the word characters.<br><br>The description of the standard type WideString has been changed to the version 1.03 description of WideCharArray with a clarification. The clarification defines the size of the WideString in terms of UTF-16 Code Units rather than the use of the words UTF-16 characters.<br><br>A summary of the changes is show in the table below.
<table>
<tr><th>Type name</th><th>Original 1.03 Description</th><th>Revised 1.03 Description</th></tr>
<tr><td>WideChar</td><td>A 16-bit UTF-16 character value</td><td>Removed</td></tr>
<tr><td>String</td><td>A null terminated sequence of UTF-8 characters</td><td>A sequence of UTF-8 characters preceded by the number of UTF-8 Code Units (bytes).</td></tr>
<tr><td>CharArray</td><td>A sequence of UTF-8 characters preceded by the number of characters</td><td>Removed</td></tr>
<tr><td>WideString</td><td>A null terminated sequence of UTF-16 characters</td><td>A sequence of UTF-16 characters preceded by the number of UTF-16 Code Units</td></tr>
<tr><td>WideCharArray</td><td>A sequence of UTF-16 characters preceded by the number of characters</td><td>Removed</td></tr>
</table></td></tr>
</table>

| | |
|---|---|
| **Topic** | Remove the concept of NamingRule |
| **Errata Version** | 1.03.9 |
| **Spec Reference** | Part 3<br>5.5.1 Object NodeClass<br>6.4.4.2 Properties describing ModellingRules<br>6.4.4.3 Subtyping Rules for Properties of ModellingRules<br>6.4.4.5.1 Titles of Standard ModellingRules<br>6.4.4.5.2 Mandatory<br>6.4.4.5.3 Optional<br>8.29 NamingRuleType |
| **Mantis Reference** | 0007909 |
| **Problem Statement** | The NamingRule concept is flawed and should be removed. |

**Solution**

In 5.5.1
Remove the row “NamingRule” from Table 6.
Remove the sentence “The Property NamingRule defines the NamingRule of a ModellingRule and shall only be applied to Objects of type ModellingRuleType.”.

Replace 6.4.4.2 and 6.4.4.3 with the following:

**6.4.4.2 Subtyping Rules for Properties of ModellingRules**

It is allowed that subtypes override ModellingRules on their InstanceDeclarations. As a general rule for subtyping, constraints shall only be tightened, not loosened. Therefore, it is not allowed to specify on the supertype that an instance shall exist with the ModellingRule Mandatory and on the subtype make this ModellingRule Optional. Table 15 specifies the allowed changes on the Properties when overriding the ModellingRules in the subtype.

**Table 15 - Rule for ModellingRules Properties when Subtyping**

| ModellingRule on supertype | ModellingRule on subtype |
|---|---|
| Mandatory | Mandatory |
| Optional | Mandatory or Optional |
| MandatoryPlaceholder | MandatoryPlaceholder |
| OptionalPlaceholder | MandatoryPlaceholder or OptionalPlaceholder |

Remove 6.4.4.5.1.

Replace the first paragraph of 6.4.4.5.2 with the following:

An *InstanceDeclaration* marked with the *ModellingRule Mandatory* means that for each existing *BrowsePath* on the instance a similar *Node* shall exist, but it is not defined whether a new *Node* is created or an existing *Node* is referenced.

Replace the first paragraph of 6.4.4.5.3 with the following:

An *InstanceDeclaration* marked with the ModellingRule *Optional* means that for each existing *BrowsePath* on the instance a similar *Node* may exist, but it is not defined whether a new *Node* is created or an existing *Node* is referenced.

Remove 8.29.

## OPC UA Specification: Part 4 – Services

| | |
|---|---|
| **Topic** | Basic128Rsa15 User Name Password encryption can be exploited. |
| **Errata Version** | 1.03.5 |
| **Spec Reference** | Part 4<br>7.36 UserIdentityToken parameters |
| **Mantis Reference** | 0004155 |
| **Problem Statement** | If a client adds unnecessary padding at the end of a user name token then the server can be used as an oracle and allow an attacker to make use of the server private key.<br><br>The issue is described CVE-2018-7559. |
| **Solution** | Add the following statements:<br><br>7.36.1 Overview (before Table 176)<br><br>To prevent the leakage of information useful to attackers, *Servers* should ensure the process of validating *UserIdentityTokens* completes in constant time whether an error occurs or not. The process of validation includes decrypting, checking padding and checking for a valid nonce. If any errors occur the return code is *Bad_IdentityTokenInvalid*.<br><br>*Servers* shall log details of any failure to validate a *UserIdentityToken* and should lock out *Client* applications with multiple failures (5 or so).<br><br>After Table 176<br><br>A *Client* should not add any padding after the secret. If a *Client* adds padding then all bytes shall be zero. A *Server* shall check for padding added by *Clients* and ensure that all padding bytes are zeros. *Servers* shall reject *UserIdentityTokens* with invalid padding. Administrators shall be able to configure *Servers* to accept *UserIdentityTokens* with invalid padding. |

| | |
|---|---|
| **Topic** | Some clients do not validate the certificate for username encryption. |
| **Errata Version** | 1.03.6 |
| **Spec Reference** | Part 4<br>7.36 UserIdentityToken |
| **Mantis Reference** | 0004231 |
| **Problem Statement** | User credentials can be compromised if a client does not validate a server certificate before sending an encrypted UserIdentityToken.<br><br>The issue is described CVE-2018-12087. |
| **Solution** | Add the following statement:<br><br>7.36 UserIdentityToken<br><br>*Clients* shall validate the *Server Certificate* and ensure it is trusted before sending a *UserIdentityToken* encrypted with the *Certificate*. |

| | |
|---|---|
| **Topic** | Order or fields in NodeAttributes Structures |
| **Errata Version** | 1.03.8 |
| **Spec Reference** | Part 4<br>7.19 NodeAttributes parameters |
| **Mantis Reference** | 0004353 |
| **Problem Statement** | The order of fields in the NodeAttributes Structures is different than in the data type schema files |
| **Solution** | 7.19 NodeAttributes parameters<br>Table 146 – ObjectAttributes<br>Table 147 – VariableAttributes<br>Table 148 – MethodAttributes<br>Table 149 – ObjectTypeAttributes<br>Table 150 – ObjectTypeAttributes<br>Table 151 – ReferenceTypeAttributes<br>Table 152 – DataTypeAttributes<br>Table 153 – ViewAttributes<br><br>Move structure fields 'writeMask' and 'userWriteMask' up in the list behind the structure field 'description' |

| | |
|---|---|
| **Topic** | Subscription retransmission queue |
| **Errata Version** | 1.03.8 |
| **Spec Reference** | Part 4<br>5.13 Subscription Service Set<br>5.13.1.1 Description<br>5.13.5 Publish |
| **Mantis Reference** | 0005634 0004795 |
| **Problem Statement** | Specification text requires retransmission queue and is not prepared for profiles who make the retransmission queue optional |
| **Solution** | 5.13 Subscription Service Set<br>5.13.1.1 Description<br>Replace i) with the following text:<br>Sessions maintain a retransmission queue of sent NotificationMessages. NotificationMessages are retained in this queue until they are acknowledged. The Session shall maintain a retransmission queue size of at least two times the number of Publish requests per Session the Server supports. A Profile in OPC 10000-7 may make the retransmission queue support optional. The minimum number of Publish requests per Session the Server shall support is defined in OPC 10000-7. Clients are required to acknowledge NotificationMessages as they are received if the Publish response parameter availableSequenceNumbers is not an empty array. An empty array in availableSequenceNumbers indicates that the Server does not support a retransmission queue and acknowledgement of NotificationMessages. In the case of a retransmission queue overflow, the oldest sent NotificationMessage gets deleted. If a Subscription is transferred to another Session, the queued NotificationMessages for this Subscription are moved from the old to the new Session.<br><br>5.13.5 Publish<br><br>Table 95 – Publish Service Parameters<br>Replace availableSequenceNumbers description with the following text:<br>A list of sequence number ranges that identify unacknowledged NotificationMessages that are available for retransmission from the Subscription's retransmission queue including the sequence number of this response if it is not a keep-alive Message. This list is prepared after processing the acknowledgements in the request (see 7.8 for Counter definition).<br>The list shall be empty if the Server does not support the retransmission queue. If the list is empty, the Client should not acknowledge sequence numbers.<br><br>Table 97 – Publish Operation Level Result Codes<br>Add status code to table<br>Good_RetransmissionQueueNotSupported<br>The Server does not support retransmission queue and acknowledgement of sequence numbers is not available. |

| | |
|---|---|
| **Topic** | Clarified StatusCode handling in Event fields |
| **Errata Version** | 1.03.8 |
| **Spec Reference** | Part 4<br>Table 115 – Basic FilterOperator definition<br>Table 118 – Conversion rules |
| **Mantis Reference** | 0004188 |
| **Problem Statement** | StatusCodes indicating not existing values in Event fields need special handling |
| **Solution** | Table 115 – Basic FilterOperator definition<br>Operator IsNull<br>Add to Description:<br>TRUE If the value in operand[0] is a StatusCode instead of the field DataType.<br><br>Table 118 – Conversion rules<br>Source Type StatusCode<br>Change all implicit conversion to explicit conversions |

## OPC UA Specification: Part 5 – Information Model

| | |
|---|---|
| **Topic** | Improve the description for the situation where collection of diagnostic information is disabled. |
| **Errata Version** | 1.03.1 |
| **Spec Reference** | Part 5<br>Clause 6.3.3 – ServerDiagnosticsType, last paragraph |
| **Mantis Reference** | 0003219 |
| **Problem Statement** | The current description is ambiguous about what nodes have to exist in the AddressSpace if the EnabledFlag is FALSE (collection disabled). |
| **Solution** | Add the following statement:<br>When diagnostics are turned off, the Server can return Bad_NodeIdUnknown for all static diagnostic *Nodes* except the *EnabledFlag Property*. Dynamic diagnostic *Nodes* (such as the *Session Nodes*) will not appear in the *AddressSpace*.<br>If collection of diagnostic information is not supported at all, the *EnabledFlag Property* will be *ReadOnly*.<br><br>It replaces the following paragraph:<br><br>> Static diagnostic *Nodes* that always appear in the *AddressSpace* will return Bad_NotReadable when the *Value Attribute* of such a *Node* is read or subscribed to and diagnostics are turned off. Dynamic diagnostic *Nodes* (such as the *Session Nodes*) will not appear in the *AddressSpace* when diagnostics are turned off. |

| | |
|---|---|
| **Topic** | *OperationLimitsType* interpretation of value=0 |
| **Errata Version** | 1.03.4 |
| **Spec Reference** | Part 5<br>6.3.11 *OperationLimitsType* first sentence after Table 18 |
| **Mantis Reference** | 0003755 |
| **Problem Statement** | The specification does not define the meaning of an operation limit which has a value of 0. |
| **Solution** | A value of 0 is not valid for operation limits. The statement “Any operational limits *Property* that is provided shall have a non-zero value” has been added |

| | |
|---|---|
| **Topic** | *SessionDiagnosticVariableType* component *BrowseNames*. |
| **Errata Version** | 1.03.4 |
| **Spec Reference** | Part 5<br>7.16 *SessionDiagnosticVariableType*, Table 75 |
| **Mantis Reference** | 0003790 |
| **Problem Statement** | There is a *BrowseName* mismatch between the specification and the generated code for two of the *Component References*. Generated code, such as the ANSI-C Stack use the *BrowseNames TotalRequestCount* and *UnauthorizedRequestCount* whereas the specification define the BrowseNames to be *TotalRequestsCount* and *UnauthorizedRequestsCount*. |
| **Solution** | The *BrowseName* used by the code generate are considered to be correct and therefore the specification has been changed to *TotalRequestCount* and *UnauthorizedRequestCount*. |

| | |
|---|---|
| **Topic** | Correction to *DataTypeDescriptionType* |
| **Errata Version** | 1.03.6 |
| **Spec Reference** | Part 5<br><br>7.7 – DataTypeDescriptionType |
| **Mantis Reference** | 0004316 |
| **Problem Statement** | The *DataType* of the *DataTypeDescriptionType* is incorrectly specifies. |
| **Solution** | The *DataType* defined in Table 66 should be *String* and will be correct in version 1.05. |

| | |
|---|---|
| **Topic** | Correction to *ServerCapabilitiesType* |
| **Errata Version** | 1.03.8 |
| **Spec Reference** | Part 5<br>6.3.2 *ServerCapabiltiesType* |
| **Mantis Reference** | 0004344 |
| **Problem Statement** | *ServerProfileArray* content clarification. |
| **Solution** | Replaced description text of ServerProfileArray after Table 10 with the following:<br>*ServerProfileArray* lists the *Profiles* that the *Server* supports. The *String* should be the *URI* of the *Profile*. See Part 7 for definitions of OPC UA *Server Profiles*. This list should be limited to the *Profiles* the *Server* supports in its current configuration.<br>Note: In Version 1.05 the String shall be the URI of the Profile. |

| | |
|---|---|
| **Topic** | *ServerType ServiceLevel Property* |
| **Errata Version** | 1.03.8 |
| **Spec Reference** | Part 5<br>6.31 ServerType |
| **Mantis Reference** | 0006257 |
| **Problem Statement** | Clarify *ServiceLevel* requirement for non-redundant *Server* |
| **Solution** | Replace the definition of the *ServiceLevel Property* which follows after Table 8 with:<br><br>*ServiceLevel* describes the ability of the *Server* to provide its data to the *Client*. The value range is from 0 to 255, where 0 indicates the worst and 255 indicates the best. Part 4 defines required sub-ranges for different scenarios. A *Server* should set the *ServiceLevel* to the most appropriate value, however if an accurate value cannot be determined the value shall be set to 255. |

| | |
|---|---|
| **Topic** | *ServerCapabilitiesType* |
| **Errata Version** | 1.03.8 |
| **Spec Reference** | Part 5<br>6.3.2 ServerCapabilitiesType |
| **Mantis Reference** | 0006040 |
| **Problem Statement** | Clarify *MaxArrayLength, MaxStringLength* and *MaxByteStringLength* are not limited to *Variables* |
| **Solution** | Replace the definition of the *MaxArrayLength Property* which follows after Table 9 with:<br><br>The MaxArrayLength Property indicates the maximum length of a one or multidimensional array supported by Variables, *Method Arguments* and *Event* fields of the Server. In a multidimensional array it indicates the overall length. For example, a three-dimensional array of 2x3x10 has the array length of 60. The Server might further restrict the length for individual Variables, *Method Arguments* or *Event* fields without notice to the client. Servers may use the Property MaxArrayLength defined in Part 3 on individual DataVariables to specify the size on individual values. The individual Property may have a larger or smaller value than MaxArrayLength.<br><br>Replace the definition of the *MaxStringLength Property* which follows after Table 9 with:<br><br>The MaxStringLength Property indicates the maximum length of Strings supported by Variables*, Method Arguments* and *Event* fields of the Server. The Server might further restrict the String length for individual Variables, *Method Arguments* or *Event* fields without notice to the client. Servers may use the Property MaxStringLength defined in Part 3 on individual DataVariables to specify the length on individual values. The individual Property may have larger or smaller values than MaxStringLength.<br><br>Replace the definition of the *MaxByteStringLength Property* which follows after Table 9 with:<br><br>The MaxByteStringLength Property indicates the maximum number of Bytes in a ByteString supported by Variables, Method Arguments, Event fields or FileType Objects of the Server. The Server might further restrict the ByteString length for individual Variables, Method Arguments, Event fields or FileType Objects without notice to the Client. Servers may use the Property MaxByteStringLength on individual DataVariables to specify the length on individual values or on FileType Objects to specify the maximum size of read and write buffers. The individual Property may have larger or smaller values than MaxByteStringLength. |

| | |
|---|---|
| **Topic** | *OptionSetType* |
| **Errata Version** | 1.03.8 |
| **Spec Reference** | Part 5<br>Table 78 OptionSetType Definition |
| **Mantis Reference** | 0006320 |
| **Problem Statement** | OptionSetType is defined with a ValueRank of Scalar and incorrectly includes the ArrayDimensions Attribute. |
| **Solution** | In Table 78 remove the ArrayDimensions Attribute. |

| | |
|---|---|
| **Topic** | *AuditEventType* |
| **Errata Version** | 1.03.8 |
| **Spec Reference** | Part 5<br>6.4.3 AuditEventType |
| **Mantis Reference** | 0006469 |
| **Problem Statement** | AuditEntryId is part of the encrypted body of the OpenSecureChannel request. All of the certificate checks are executed before the body is decrypted. If one of the certificate checks fails, decrypt of the body does not take place. The expected content of ClientAuditEntryId is not described for this error condition. |
| **Solution** | In 6.4.3 replace the description of ClientAuditEntryId with<br>*ClientAuditEntryId* contains the human-readable *AuditEntryId* defined in OPC 10000-4 If the Server is unable to decrypt AuditEntryId due to a certificate check failure, then the Client's IP Address shall be used as the ClientAuditEntryId. |

<table>
<tr><td>Topic</td><td>WssIdentityToken should be IssuedIdentityToken</td></tr>
<tr><td>Errata Version</td><td>1.03.8</td></tr>
<tr><td>Spec Reference</td><td>Part 5<br>12.3 DataTypes defined in Part 5</td></tr>
<tr><td>Mantis Reference</td><td>0006740</td></tr>
<tr><td>Problem Statement</td><td>The name of IssuedIdentityToken was incorrectly stated as WssIdentityToken.</td></tr>
<tr><td>Solution</td><td>In 12.3 replace Table 131 with:
<table>
<tr><th>BrowseName</th></tr>
<tr><td>AnonymousIdentityToken</td></tr>
<tr><td>DataValue</td></tr>
<tr><td>DiagnosticInfo</td></tr>
<tr><td>ExpandedNodeId</td></tr>
<tr><td>SignedSoftwareCertificate</td></tr>
<tr><td>UserIdentityToken</td></tr>
<tr><td>UserNameIdentityToken</td></tr>
<tr><td>X509IdentityToken</td></tr>
<tr><td>IssuedIdentityToken</td></tr>
<tr><td>SecurityTokenRequestType</td></tr>
<tr><td>AddNodesItem</td></tr>
<tr><td>AddReferencesItem</td></tr>
<tr><td>DeleteNodesItem</td></tr>
<tr><td>DeleteReferencesItem</td></tr>
<tr><td>NumericRange</td></tr>
<tr><td>MessageSecurityMode</td></tr>
<tr><td>ApplicationDescription</td></tr>
</table>
In 12.3 replace Table 132 with:
<table>
<tr><th>Attributes</th><th colspan="3">Value</th></tr>
<tr><td>BrowseName</td><td colspan="3">UserIdentityToken</td></tr>
<tr><td>IsAbstract</td><td colspan="3">TRUE</td></tr>
<tr><th>References</th><th>NodeClass</th><th>BrowseName</th><th>IsAbstract</th></tr>
<tr><td>HasSubtype</td><td>DataType</td><td>UserNameIdentityToken</td><td>FALSE</td></tr>
<tr><td>HasSubtype</td><td>DataType</td><td>X509IdentityToken</td><td>FALSE</td></tr>
<tr><td>HasSubtype</td><td>DataType</td><td>IssuedIdentityToken</td><td>FALSE</td></tr>
<tr><td>HasSubtype</td><td>DataType</td><td>AnonymousIdentityToken</td><td>FALSE</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>Topic</td><td>StateMachine HasEffect ReferenceType</td></tr>
<tr><td>Errata Version</td><td>1.03.9</td></tr>
<tr><td>Spec Reference</td><td>Part 5<br>B4.14 HasEffect</td></tr>
<tr><td>Mantis Reference</td><td>7254</td></tr>
<tr><td>Problem Statement</td><td>The StateMachine annex defines the HasEffect ReferenceType. It states that it can be used on EventTypes (Transitions pointing to EventTypes). However, it is not clearly defined whether a Transition having an Event as Effect shall generate such Event every time it is triggered.</td></tr>
<tr><td>Solution</td><td>Replace B4.14 with the following:<br><br>The HasEffect ReferenceType is a concrete ReferenceType and can be used directly. It is a subtype of NonHierarchicalReferences.<br><br>The semantic of this ReferenceType is to point from a Transition to something that will be effected when the Transition is triggered. In this annex we only define EventTypes as Effects. However, the ReferenceType is not restricted to point to EventTypes.<br><br>The SourceNode of this ReferenceType shall be an Object of the ObjectType TransitionType or one of its subtypes. The TargetNode can be of any NodeClass.<br><br>If the TargetNode is an EventType, each time the Transition is triggered (either by a Client or internally in the Server) an Event of that EventType or a subtype shall be generated.<br><br>The representation of the HasEffect ReferenceType in the AddressSpace is specified in B.13.<br><br>Table B.13 – HasEffect ReferenceType
<table>
<tr><th>Attributes</th><th colspan="3">Value</th></tr>
<tr><td>BrowseName</td><td colspan="3">HasEffect</td></tr>
<tr><td>InverseName</td><td colspan="3">MayBeEffectedBy</td></tr>
<tr><td>Symmetric</td><td colspan="3">False</td></tr>
<tr><td>IsAbstract</td><td colspan="3">False</td></tr>
<tr><th>References</th><th>NodeClass</th><th>BrowseName</th><th>Comment</th></tr>
<tr><td></td><td></td><td></td><td></td></tr>
</table>
</td></tr>
</table>

| | |
|---|---|
| **Topic** | *Special Restrictions on subtyping StateMachines* |
| **Errata Version** | 1.03.9 |
| **Spec Reference** | Part 5<br>B.4.18 Special Restrictions on subtyping StateMachines |
| **Mantis Reference** | 0005683 |
| **Problem Statement** | Clarification on Subtyping StateMachines needed |
| **Solution** | Replace B.4.18 with the following:<br><br>In general, all rules on subtyping apply for *StateMachine* types as well. Some additional rules apply for *StateMachine* types.<br><br>*States* and *Transitions* are not instantiated, this information is only provided on the *ObjectType*. They have no *ModellingRule*, and thus, also the inheritance of *States* and *Transitions* is not defined. Therefore, the following rules apply for subtyping *StateMachines*. Each *State* and *Transition* defined on the supertype shall be available on the subtype as well. That is, for each *State* defined on the supertype another *Node* of the same *ObjectType* having the same *BrowseName* and the same *StateNumber* shall be defined on the subtype. For each *Transition* defined on the supertype another *Node* of the same *ObjectType* having the same *BrowseName* and the same *TransitionNumber* shall be defined on the subtype. All references defining the *StateMachine* (*HasCause*, *HasEffect*, *FromState*, *ToState*, *HasSubStateMachine*) shall be replicated in the subtype as well. If *InstanceDeclarations* are referenced (e.g., *Methods* used to trigger *Transitions*) either the *InstanceDeclaration* of the supertype is referenced or the *InstanceDeclaration* is overridden, and in the latter case the overridden *InstanceDeclaration* of the subtype shall be referenced.<br><br>If a StateMachine type is not abstract, subtypes of it shall not change the behaviour of it. That means, that in this case a subtype shall not add *States* and it shall not add *Transitions* between its *States*. However, a subtype may add *SubStateMachines*, it may add *Transitions* from the *States* to the *States* of the *SubStateMachine*, and it may add *Causes* and *Effects* to a *Transition*. In addition, a subtype of a *StateMachine* type shall not remove *States* or *Transitions*. |

| | |
|---|---|
| **Topic** | *ModellingRules for States and Transitions* |
| **Errata Version** | 1.03.9 |
| **Spec Reference** | Part 5<br>B.4.8 StateType<br>B.4.10 TransitionType |
| **Mantis Reference** | 0007761 |
| **Problem Statement** | Clarification on ModellingRules for States and Transitions in StateMachines needed |
| **Solution** | In B.4.8 replace the first paragraph with the following:<br>*States* of a *FiniteStateMachine* are represented as *Objects* of the *StateType.* Each *Object* of the *StateType* or one of its subtypes shall be referenced from the *ObjectType FiniteStateMachineType* or one of its subtypes using a *HasComponent Reference* or a subtype of *HasComponent* and shall not have a *ModellingRule* as they are not applied on the instances.<br>In B.4.10 replace the first paragraph with the following:<br>*Transitions* of a *FiniteStateMachine* are represented as *Objects* of the *ObjectType TransitionType* formally defined in Table B.9. Each *Object* of the *TransitionType* or one of its subtypes shall be referenced from the *ObjectType FiniteStateMachineType* or one of its subtypes using a *HasComponent Reference* or a subtype of *HasComponent* and shall not have a *ModellingRule* as they are not applied on the instances. |

## OPC UA Specification: Part 6 – Mappings

| | |
|---|---|
| **Topic** | Fixes inconsistencies between NodeSet and specification. |
| **Errata Version** | 1.03.2 |
| **Spec Reference** | Part 6<br>Annex B OPC UA Nodeset |
| **Mantis Reference** | 0003289, 0003298, 0003296, 0002655, 0003295, 0003308, 0003294, 0003309 |
| **Problem Statement** | The Nodeset has several errors that need to be fixed. |
| **Solution** | The InverseName is now not specified for the References ReferenceType (Mantis #3289);<br><br>The InverseName of HasSubtype is now SubtypeOf instead of HasSupertype (Mantis #3298);<br><br>The SessionPlaceholder component of in SessionsDiagnosticsSummaryType has been renamed to ClientName (Mantis #3296);<br><br>OperationLimitsType is a subtype of FolderType instead of BaseObjectType (Mantis #2655);<br><br>StaticNodeIdIdentifierTypes of NamespaceMetadataType has been renamed to StaticNodeIdTypes (Mantis #3295);<br><br>An instance of NamespaceMetadataType for the OPCUANamespace has been added (Mantis #3308);<br><br>Added ExpirationLimit to CertificateExpirationAlarmType (Mantis #3420);<br><br>CertificateGroup component of CertificateGroupFolderType has been renamed to AdditionalGroup (Mantis #3294);<br><br>StaticStringNodeIdPattern ValueRank is Scalar instead of Array (Mantis #3309) |

| | |
|---|---|
| **Topic** | URL links to Supporting Files should say 1.03 - not 1.02. |
| **Errata Version** | 1.03.2 |
| **Spec Reference** | Part 6<br>Annex A through F |
| **Mantis Reference** | 0003150 |
| **Problem Statement** | The URLs in Annex A through F are incorrect. |
| **Solution** | http://www.opcfoundation.org/UA/schemas/1.03/StatusCode.csv<br>http://www.opcfoundation.org/UA/schemas/1.03/NodeIds.csv<br>http://www.opcfoundation.org/UA/schemas/1.03/Opc.Ua.NodeSet2.xml<br>http://www.opcfoundation.org/UA/schemas/1.03/Opc.Ua.Types.bsd.xml<br>http://www.opcfoundation.org/UA/schemas/1.03/Opc.Ua.Endpoints.wsdl<br>http://www.opcfoundation.org/UA/schemas/1.03/SecuredApplication.xsd<br>http://www.opcfoundation.org/UA/schemas/1.03/UANodeSet.xsd |

| | |
|---|---|
| **Topic** | Inconsistency between NodeSet Schema and Part 6. |
| **Errata Version** | 1.03.2 |
| **Spec Reference** | Part 6<br>Annex F Information Model XML Schema |
| **Mantis Reference** | 0003184 |
| **Problem Statement** | The Version attribute in "UANodeSet" has been replaced by LastModified but the Version attribute was not removed. |
| **Solution** | Removed the Version from the UANodeSet Schema and posted the updated schema here:<br>http://www.opcfoundation.org/UA/schemas/1.03/UANodeSet.xsd |

| | |
|---|---|
| **Topic** | Symmetric ReferenceTypes shall not have the InverseName. |
| **Errata Version** | 1.03.2 |
| **Spec Reference** | Part 6<br>Annex F Information Model XML Schema |
| **Mantis Reference** | 0003289 |
| **Problem Statement** | UAReferenceType NodeId="i=31" has an InverseName.<br>Code Generator needs to be fixed. |
| **Solution** | Removed the InverseNames for all Symmetric types. Updated NodeSet here:<br><br>http://www.opcfoundation.org/UA/schemas/1.03/UANodeSet.xsd |

| | |
|---|---|
| **Topic** | Fixes inconsistencies between NodeSet and specification. |
| **Errata Version** | 1.03.2 |
| **Spec Reference** | Part 6<br>Annex F Information Model XML Schema |
| **Mantis Reference** | 00003292 |
| **Problem Statement** | FiniteStateMachineType uses the default for IsAbstract which is false.<br>It should be true. |
| **Solution** | FiniteStateMachineType should have IsAbstract=true. |

| | |
|---|---|
| **Topic** | New license related error codes added to Part 4 in 1.03 are not in code files. |
| **Errata Version** | 1.03.2 |
| **Spec Reference** | Part 6<br>Annex A.2 StatusCodes |
| **Mantis Reference** | 00003297 |
| **Problem Statement** | StatusCodes.csv is missing the new StatusCodes. |
| **Solution** | Re-generated file. Posted update here:<br>http://www.opcfoundation.org/UA/schemas/1.03/StatusCode.csv |

| | |
|---|---|
| **Topic** | Issues with encoding mask length for structures with optional fields. |
| **Errata Version** | 1.03.2 |
| **Spec Reference** | Part 6<br>5.2.6 Structures with optional fields |
| **Mantis Reference** | 00003426 |
| **Problem Statement** | Implementations have issues with variable length encoding masks because there is no way to publish the length. |
| **Solution** | Changed 5.2.6 to:<br><br>The *EncodingMask* is a 32-bit unsigned integer. Each optional field is assigned exactly one bit, however, a single bit may control multiple fields.<br><br>Also deleted text that explains how to calculate the mask length. |

| | |
|---|---|
| **Topic** | Definition of min send and receive buffer size wrong. |
| **Errata Version** | 1.03.3 |
| **Spec Reference** | Part 6<br>6.7.2 MessageChunk structure and<br>7.1.2 MessageChunk structure |
| **Mantis Reference** | 0003447 |
| **Problem Statement** | The specification specifies a minimum message size of 8196 for UA Secure Conversation and states that the message size is greater than 8192 for UA TCP. The layers are independent so different values are not wrong but they are confusing. |
| **Solution** | Remove the limits from the OPC UA TCP description.<br>Change the limit to >= 8192 for OPC UA Secure Conversation. |

| | |
|---|---|
| **Topic** | Fixes inconsistencies between NodeSet and specification. |
| **Errata Version** | 1.03.4 |
| **Spec Reference** | Part 6<br>Annex B OPC UA NodeSet |
| **Mantis Reference** | 0003064, 0003466, 0003468, 0003472, 0003509, 0003510 |
| **Problem Statement** | The NodeSet has several errors that need to be fixed. These changes do not affect the specification. All changes are published the of 1.03 branch of the UA-NodeSet repository which can be found here: https://github.com/OPCFoundation/UA-Nodeset/tree/v1.03 |
| **Solution** | Integer and UInteger are now direct subtypes of Number as described in Part 3 (Mantis# 3064 and 2526);<br><br>Adopted a consistent naming convention for instances with the Mandatory Placeholder or OptionalPlaceholder ModellingRule (Mantis #3466);<br><br>Removed KerberosIdentityToken because it is not in the specification (Mantis #3468);<br><br>AlwaysGeneratesEvent is now a subtype of GeneratesEvent (Mantis #3472);<br><br>Added Invalid to List of Values for OpcUa_BrowseDirection and OpcUa_TimestampsToReturn (Mantis #3509);<br><br>Removed SoftwareCertificate because it is not defined in specification (Mantis #3510); |

| | |
|---|---|
| **Topic** | HTTPS default Port should be 443. |
| **Errata Version** | 1.03.4 |
| **Spec Reference** | Part 6<br>7.6 Well known addresses |
| **Mantis Reference** | 0003497 |
| **Problem Statement** | HTTPS default Port is 4843 in Part 6. The normal use-case for HTTPS is when traffic is limited by IT departments to HTTP over standard ports. |
| **Solution** | Changed default LDS HTTPS URL port from 4843 to 443. |

| | |
|---|---|
| **Topic** | UA Binary encoding for Structures with optional fields needs clarification. |
| **Errata Version** | 1.03.4 |
| **Spec Reference** | Part 6<br>5.2.6 Structures with optional fields |
| **Mantis Reference** | 0003574 |
| **Problem Statement** | The specification allowed a single bit to control multiple fields. This is not supported by the UANodeSet schema. |
| **Solution** | Removed text “however, a single bit may control multiple fields.” |

| | |
|---|---|
| **Topic** | Handling of NaN. |
| **Errata Version** | 1.03.4 |
| **Spec Reference** | Part 6<br>5.2.2.3 Floating Point |
| **Mantis Reference** | 0003575 |
| **Problem Statement** | The specification required that NaNs always produce data changes. This makes NaN unusable in practice. |
| **Solution** | Removed:<br>“This means a NaN value for a Variable always produces a *DataChange* each time the SamplingInterval elapses.” |

| | |
|---|---|
| **Topic** | Encoding of empty Variant is not explicitly specified. |
| **Errata Version** | 1.03.4 |
| **Spec Reference** | Part 6<br>5.2.2.16 Variant |
| **Mantis Reference** | 0003709 |
| **Problem Statement** | The Encoding of empty Variant is not explicitly specified. This is an oversight. All implementations handle this case. |
| **Solution** | Added to description of EncodingMask:<br>“A value of 0 specifies a NULL and that no other fields are encoded.” |

| | |
|---|---|
| **Topic** | Rules for DataTypes allowed in Variant. |
| **Errata Version** | 1.03.4 |
| **Spec Reference** | Part 6<br>5.1.6 Variant |
| **Mantis Reference** | 0003718 |
| **Problem Statement** | The text currently prohibits any use of DataValue in a Variant. It should limit the restriction to Values of Attributes. |
| **Solution** | Changed text from:<br><br>DataValue and DiagnosticInfo types only have meaning when returned in a response message with an associated StatusCode. As a result, Variants cannot contain instances of DataValue or DiagnosticInfo.<br><br>to:<br><br>*Values* of *Attributes* are always returned in instances of *DataValues*. Therefore, the *DataType* of an *Attribute* cannot be a *DataValue*. *Variants* can contain *DataValue* when used in other contexts such as *Method Arguments* or *PubSub Messages*. The *Variant* in a *DataValue* cannot, directly or indirectly, contain another *DataValue*. |

| | |
|---|---|
| **Topic** | Missing optional EncodingMask position information. |
| **Errata Version** | 1.03.4 |
| **Spec Reference** | Part 6<br>5.3.5 Structures with optional fields. |
| **Mantis Reference** | 0003776 |
| **Problem Statement** | There is no way to know which bits are assigned to which fields in 5.3.5, 5.2.6 and 5.4.6 |
| **Solution** | Changed text from:<br><br>The *EncodingMask* is a 32-bit unsigned integer. Each optional field is assigned exactly one bit. The bits assigned to fields may not be contiguous. Unassigned bits are set to 0 by encoders. Decoders shall report an error if assigned bits are not 0.<br><br>to:<br><br>The *EncodingMask* is a 32-bit unsigned integer. Each optional field is assigned exactly one bit. The bits assigned to fields are contiguous and are assigned in the order they appear in the *Structure*. Unassigned bits are set to 0 by encoders. Decoders shall report an error if unassigned bits are not 0. |

| | |
|---|---|
| **Topic** | MaxBodySize formula error. |
| **Errata Version** | 1.03.4 |
| **Spec Reference** | Part 6<br>Table 30 – OPC UA Secure Conversation Message footer |
| **Mantis Reference** | 0002897 |
| **Problem Statement** | The formula for cacalculating MaxBodySize is incorrect. |
| **Solution** | Changed text from:<br>`MaxBodySize = PlainTextBlockSize * Floor((MessageChunkSize - HeaderSize - SignatureSize - 1) / CipherTextBlockSize) - SequenceHeaderSize;`<br>to:<br>`MaxBodySize = PlainTextBlockSize * Floor((MessageChunkSize - HeaderSize) / CipherTextBlockSize) - SequenceHeaderSize - SignatureSize - PaddingByteSize;` |

| | |
|---|---|
| **Topic** | AuditConditionCommentEventType Property should be Named ConditionEventId |
| **Errata Version** | 1.03.8 |
| **Spec Reference** | TargetVersion="1.03.8"<br>TargetPublicationDate="2021-07-15" |
| **Mantis Reference** | 0004818 |
| **Problem Statement** | AuditConditionCommentEventType Property should be Named ConditionEventId. |
| **Solution** | Rename:<br>EventId => ConditionEventId in<br>AuditConditionCommentEventType<br>AuditConditionAcknowledgeEventType<br>AuditConditionConfirmEventType. |

| | |
|---|---|
| **Topic** | Wrong structure of ServerRedundancy node in Opc.Ua.NodeSet2.xml |
| **Errata Version** | 1.03.8 |
| **Spec Reference** | TargetVersion="1.03.8"<br>TargetPublicationDate="2021-03-10" |
| **Mantis Reference** | 0006232 |
| **Problem Statement** | The ServerRedundancy node defines well-known Nodes for all Properties of all possible subtypes. This is confusing to users. |
| **Solution** | The references from the ServerRedundancy node to the well-known Property nodes for subtypes have been deleted.<br>The Properties are still defined as unattached Nodes. |

| | |
|---|---|
| **Topic** | Wrong BrowseName for EventQueueOverflowCount |
| **Errata Version** | 1.03.8 |
| **Spec Reference** | TargetVersion="1.03.8"<br>TargetPublicationDate="2021-03-10" |
| **Mantis Reference** | 0003909 |
| **Problem Statement** | Wrong BrowseName for EventQueueOverflowCount. |
| **Solution** | Change EventQueueOverFlowCount to EventQueueOverflowCount. |

| | |
|---|---|
| **Topic** | Properties on DataTypes have ModelingRules. |
| **Errata Version** | 1.03.8 |
| **Spec Reference** | TargetVersion="1.03.8"<br>TargetPublicationDate="2021-07-15" |
| **Mantis Reference** | 0007052 |
| **Problem Statement** | Properties on DataTypes have ModelingRules that should not be there. |
| **Solution** | Removed ModellingRules on all Properties of DataTypes, RefrenceTypes and Views. |

| | |
|---|---|
| **Topic** | DiagnosticInfo has wrong length for encoding mask. |
| **Errata Version** | 1.03.8 |
| **Spec Reference** | TargetVersion="1.03.8"<br>TargetPublicationDate="2021-07-15" |
| **Mantis Reference** | 0003252 |
| **Problem Statement** | DiagnosticInfo has wrong length for encoding mask. |
| **Solution** | Updated Opc.Ua.Types.bsd changed Length to 1:<br><br><opc:Field Name="Reserved1" TypeName="opc:Bit" Length="1"/> |

| | |
|---|---|
| **Topic** | SystemOffNormalAlarmType IsAbstract should be False. |
| **Errata Version** | 1.03.8 |
| **Spec Reference** | TargetVersion="1.03.8"<br>TargetPublicationDate="2021-07-15" |
| **Mantis Reference** | 0006997 |
| **Problem Statement** | SystemOffNormalAlarmType IsAbstract should be False. |
| **Solution** | Change SystemOffNormalAlarmType IsAbstract to False. |

| | |
|---|---|
| **Topic** | Wrong BrowserName for Server_ServerRedundancy components. |
| **Errata Version** | 1.03.8 |
| **Spec Reference** | TargetVersion="1.03.8"<br>TargetPublicationDate="2021-07-15" |
| **Mantis Reference** | 0006549 |
| **Problem Statement** | Wrong BrowserName for Server_ServerRedundancy components. |
| **Solution** | Fix BrowseName for:<br>Server_ServerRedundancy_CurrentServerId<br>Server_ServerRedundancy_RedundantServerArray<br>Server_ServerRedundancy_ServerUrl<br>Server_ServerRedundancy_ServerNetworkGroupsiArray |

| | |
|---|---|
| **Topic** | EnumValues and Enum DataTypeDefinitions missing Descriptions. |
| **Errata Version** | 1.03.9 |
| **Spec Reference** | TargetVersion="1.03.9"<br>TargetPublicationDate="2022-03-29" |
| **Mantis Reference** | 0007901 |
| **Problem Statement** | EnumValues and Enum DataTypeDefinitions missing Descriptions. |
| **Solution** | Add descriptions to every EnumValueType instance in NodeSet. |

| | |
|---|---|
| **Topic** | Clarification on NamingRule. |
| **Errata Version** | 1.03.9 |
| **Spec Reference** | TargetVersion="1.03.9"<br>TargetPublicationDate="2022-03-29" |
| **Mantis Reference** | 0007902 |
| **Problem Statement** | NamingRule is obsolete and no longer serves any purpose. |
| **Solution** | Remove.NamingRule Property from all ModellingRules. |

| | |
|---|---|
| **Topic** | No way to specify XML schema namespace in NodeSet. |
| **Errata Version** | 1.03.9 |
| **Spec Reference** | TargetVersion="1.03.9"<br>TargetPublicationDate="2022-03-29" |
| **Mantis Reference** | 0007903 |
| **Problem Statement** | No way to specify XML schema namespace in NodeSet. |
| **Solution** | Add XmlSchemaUri ModelTableEntry element in UANodeSet.xsd<br><br>Updated files are here:<br>https://files.opcfoundation.org/schemas/UA/1.04/ |

### OPC UA Specification: Part 7 – Profiles

| | |
|---|---|
| **Topic** | Wrong URI for AsymmetricSignatureAlgorithm – Rsa_Sha256 |
| **Errata Version** | 1.03.4 |
| **Spec Reference** | Part 7<br>Clause 5.3 – Table 11<br>ConformanceUnit: “Security Basic 256 Sha256” |
| **Mantis Reference** | 0003601 |
| **Problem Statement** | The ConformanceUnit currently lists<br>http://www.w3.org/2001/04/xmldsig#rsa-sha256<br>but the correct URI is<br>http://www.w3.org/2001/04/xmldsig-more#rsa-sha256<br><br>Reference:<br>https://www.w3.org/TR/xmlsec-algorithms/<br><br>Applications that use different URIs cannot interoperate. |
| **Solution** | Since most implementations today already use the correct URI (http://www.w3.org/2001/04/xmldsig-more#rsa-sha256), the URI in this ConformanceUnit will be updates as well. |

| | |
|---|---|
| **Topic** | Embedded UA Server Profile requires X509 User Token |
| **Errata Version** | 1.03.7 |
| **Spec Reference** | Part 7<br>Clause 6.5.54 – Table 76<br>Embedded UA Server Profile |
| **Mantis Reference** | 0005018 |
| **Problem Statement** | Embedded servers rarely implement this user token type. Such servers cannot be certified for the V1.03 Embedded Server Profile.<br><br>In OPC UA V1.04 the support is already changed to optional. |
| **Solution** | The requirement to support the X509 user token facet is removed from the Embedded UA Server Profile. It is still required for Standard UA Server. |

| | |
|---|---|
| **Topic** | Core Client Facet Profile requires X509 User Token |
| **Errata Version** | 1.03.7 |
| **Spec Reference** | Part 7<br>Clause 6.5.58 – Table 80<br>Core Client Facet |
| **Mantis Reference** | 0005100 |
| **Problem Statement** | The X509 user token is not required for embedded UA Servers. Since the Core Client Facet should only mandate features that are essential for basic UA communication this user token shall not be mandated in this facet.<br><br>In OPC UA V1.04 the support is already changed to optional. |
| **Solution** | The requirement to support the X509 user token facet is removed from the Core Client Facet. It is available as optional facet. |

| | |
|---|---|
| **Topic** | Encrypted channel for Audit Events |
| **Errata Version** | 1.03.8 |
| **Spec Reference** | Part 7<br>Clause 6.5.31 – Table 53<br>Auditing Server Facet |
| **Mantis Reference** | 0004960 |
| **Problem Statement** | The information in an audit record may contain sensitive or private information. The Auditing Server Facet has no restrictions for sending Audit Events. |
| **Solution** | Add the following conformance unit to auditing:<br>Auditing Secure Communication - Security related Audit Events shall be provided over an encrypted channel. An encrypted channel can be an OPC UA Secure Conversation with encryption enabled or a transport that includes encryption (like IPSec). |

## OPC UA Specification: Part 8 – DataAccess

| | |
|---|---|
| **Topic** | MultiStateValueDiscreteType is inconsistent with NodeSet |
| **Errata Version** | 1.03.8 |
| **Spec Reference** | Table 6 “MultiStateValueDiscreteType definition” |
| **Mantis Reference** | 0005927 |
| **Problem Statement** | The definition of *MultiStateValueDiscreteType* defines a ValueRank of "Scalar" but the NodeSet defines "Any". |
| **Solution** | Change *ValueRank* from Scalar to "Any".<br>Add the following sentence to the description of properties below the table:<br>“If the item contains an array then the *EnumValues Property* shall apply to all elements in the array.” |

| | |
|---|---|
| **Topic** | Unclear behaviour if ValuePrecision contains negative values |
| **Errata Version** | 1.03.9 |
| **Spec Reference** | Clause 5.3.1 DataItem Type |
| **Mantis Reference** | 0007206 |
| **Problem Statement** | The description does not specify the behaviour when *ValuePrecision* has a negative value. It also recommends “rounding” but does not recommend a rounding algorithm. |
| **Solution** | Change the definition of *ValuePrecision* to the following:<br>*ValuePrecision* specifies the maximum precision that the *Server* can maintain for the item based on restrictions in the target environment.<br><br>*ValuePrecision* can be used for the following *DataTypes*:<br><br>• For Float, Double, and Decimal values it specifies the number of digits after the decimal place when it is a positive number. When it is a negative number, it specifies the number of insignificant digits to the left of the decimal place.<br>For example, a *ValuePrecision* of -2 specifies that the precision of the *Value* is to the nearest 100. The *ValuePrecision* should always be a whole number and it shall always be interpreted as a whole number by rounding it to the nearest whole number.<br>• For DateTime values it shall always be a positive number which indicates the minimum time difference in nanoseconds. For example, a *ValuePrecision* of 20 000 000 defines a precision of 20 ms. The *ValuePrecision* should always be a whole number and it shall always be interpreted as a whole number by rounding it to the nearest whole number.<br>• *ValuePrecision* can also be used for other subtypes of Double (like Duration) and other Number subtypes that can be represented by a Double.<br><br>The *ValuePrecision Property* is an approximation that is intended to provide guidance to a *Client*. A *Server* is expected to silently round any value with more precision that it supports. This implies that a *Client* may encounter cases where the value read back from a *Server* differs from the value that it wrote to the *Server*. This difference shall be no more than the difference suggested by this *Property*.<br><br>The algorithm for rounding should follow the so-called “Banker’s rounding” (aka Round half to even), in which numbers which are equidistant from the two nearest integers are rounded to the nearest even integer. Thus, 0.5 rounds down to 0; 1.5 rounds up to 2.<br><br>Other decimal fractions round as you would expect--0.4 to 0, 0.6 to 1, 1.4 to 1, 1.6 to 2, etc. Only x.5 numbers get the "special" treatment. |

## OPC UA Specification: Part 9 – Alarms & Conditions

| | |
|---|---|
| **Topic** | Error code for Confirm method is incorrect |
| **Errata Version** | 1.03.8 |
| **Spec Reference** | Part 9<br>5.7.4 Confirm Method<br>Table 31 Confirm result Codes |
| **Mantis Reference** | 0005544 |
| **Problem Statement** | The error code for the confirm method is incorrect-it includes "Bad_NodeIdUnknown" but should be Bad_NodeIdInvalid |
| **Solution** | Replace “Bad_NodeIdUnknown" with “Bad_NodeIdInvalid” in the table. |

| | | |
|---|---|---|
| **Topic** | Clarification on abstractness of LimitAlarmType | |
| **Errata Version** | 1.03.8 | |
| **Spec Reference** | Part 9<br>5.8.4 LimitAlarmType | |
| **Mantis Reference** | 0004273 | |
| **Problem Statement** | The text describe the alarm type as Abstract, but the Table definition list the alarm type as not abstract. . | |
| **Solution** | Replace<br>“The *LimitAlarmType* is an abstract type used to provide a base *Type* for *AlarmConditions* with multiple limits.”<br>With<br>“The *LimitAlarmType* is used to provide a base *Type* for *AlarmConditionTypes* with multiple limits” | |

| | |
|---|---|
| **Topic** | SystemOffNormalAlarmType was intended for direct use but IsAbstract is True |
| **Errata Version** | 1.03.8 |
| **Spec Reference** | Part 9<br>Table 56 – SystemOffNormalAlarmType definition. |
| **Mantis Reference** | 0006408 |
| **Problem Statement** | The SystemOffNormalAlarmType is intend for direct use, but is marked as abstract, it should not be. |
| **Solution** | Change IsAbstract value in Table from True to False<br>Also updated the NodeSet to match |

<table>
<tr><td>Topic</td><td>ConditionType::ConditionRefresh fail if the Subscription has no event monitored items</td></tr>
<tr><td>Errata Version</td><td>1.03.8</td></tr>
<tr><td>Spec Reference</td><td>Part 9<br>Table 17 – ConditionRefresh result codes</td></tr>
<tr><td>Mantis Reference</td><td>0006158</td></tr>
<tr><td>Problem Statement</td><td>Need to define an appropriate error code for a refresh call on a subscription that does not have an event monitored Item</td></tr>
<tr><td>Solution</td><td>Added following line to table<br><table><tr><td>Bad_NothingToDo</td><td>The ConditionRefresh Method was called on a SubscriptionId that has no Notifier MonitoredItems.</td></tr></table></td></tr>
</table>

| | |
|---|---|
| **Topic** | Clarification OneShotShelving & TimedShelving |
| **Errata Version** | 1.03.8 |
| **Spec Reference** | Part 9<br>5.8.3 ShelvedStateMachineType<br>5.8.3.1 Overview |
| **Mantis Reference** | 0004385 |
| **Problem Statement** | The behaviour of the ShelvedStateMachine needs to describe what is expected when an alarm is not active. |
| **Solution** | Replace the following paragraphs in the overview :<br>"In *OneShotShelving*, a user requests that an *Alarm* be Shelved for its current *Active* state. This type of *Shelving* is typically used when an *Alarm* is continually occurring on a boundary (i.e. a *Condition* is jumping between High *Alarm* and HighHigh *Alarm*, always in the *Active* state). The One Shot *Shelving* will automatically clear when an *Alarm* returns to an inactive state. Another use for this type of *Shelving* is for a plant area that is shutdown i.e. a long running *Alarm* such as a low level *Alarm* for a tank that is not in use. When the tank starts operation again the *Shelving* state will automatically clear.<br>In *TimedShelving*, a user specifies that an *Alarm* be shelved for a fixed time period. This type of *Shelving* is quite often used to block nuisance *Alarms*. For example, an *Alarm* that occurs more than 10 times in a minute may get shelved for a few minutes."<br>With the following:<br>"In *OneShotShelving*, a user requests that an *Alarm* be *Shelved* for its current *Active* state or if not *Active* its next *Active* state. This type of *Shelving* is typically used when an *Alarm* is continually occurring on a boundary (i.e. a *Condition* is jumping between High *Alarm* and HighHigh *Alarm*, always in the *Active* state). The *OneShotShelving* will automatically clear when an *Alarm* returns to an inactive state. Another use for this type of *Shelving* is for a plant area that is shutdown i.e. a long running *Alarm* such as a low level *Alarm* for a tank that is not in use. When the tank starts operation again the *Shelving* state will automatically clear.<br>In *TimedShelving*, a user specifies that an *Alarm* be shelved for a fixed time period. This type of *Shelving* is quite often used to block nuisance *Alarms*. For example, an *Alarm* that occurs more than 10 times in a minute may get shelved for a few minutes. The Alarm is shelved for the time period, no matter how many transitions the *Alarm* has between *Active* state and *Inactive* state." |

| | |
|---|---|
| **Topic** | Clarification needed for TwoStateVariableType TrueState and FalseState |
| **Errata Version** | 1.03.9 |
| **Spec Reference** | Part 9<br>Section 5.2 Two-state state machines.<br>Also sections: 5.5.2, 5.6.2, 5.7.2, 5.8.2, 5.8.6 (see table references below) |
| **Mantis Reference** | 0006412 |
| **Problem Statement** | TrueState & FalseState variable in TwoStateVariableType , should only exist on instance declaration, they should not be on alarm instances |

**Solution**

Replace the paragraph **following this one** –
"Other optional *Properties* of the *StateVariableType* have no defined meaning for *TwoStateVariableType*."
With the following text
*"TrueState* and *FalseState* contain the localized string for the *TwoStateVariableType* value when its *Id Property* has the value True or False, respectively. Since the two *Properties* provide meta-data for the *Type*, *Servers* shall not allow these *Properties* to be selected in the *Event* filter for a MonitoredItem. The *TrueState Property* and *FalseState Property* shall only exist on *InstanceDeclarations*. *Clients* can use the Read *Service* to get the values of the *TrueState* and *FalseState Property*. "

Also the follow additional tables were added as noted:
After Table 7 - ConditionType definition

**Table 7.a – ConditionType Additional Subcomponents**

| BrowsePath | References | NodeClass | BrowseName | DataType | TypeDefinition | Others |
|---|---|---|---|---|---|---|
| EnabledState | HasProperty | Variable | TrueState | LocalizedText | PropertyType | X |
| EnabledState | HasProperty | Variable | FalseState | LocalizedText | PropertyType | X |

After Table 22 – DialogConditionType definition

**Table 22.a– DialogConditionType Additional Subcomponents**

| BrowsePath | References | NodeClass | BrowseName | DataType | TypeDefinition | Others |
|---|---|---|---|---|---|---|
| DialogState | HasProperty | Variable | TrueState | LocalizedText | PropertyType | X |
| DialogState | HasProperty | Variable | FalseState | LocalizedText | PropertyType | X |

After Table 26 - AcknowledgeableConditionType definition

**Table 26.a – AcknowledgeableConditionType Additional Subcomponents**

| BrowsePath | References | NodeClass | BrowseName | DataType | TypeDefinition | Others |
|---|---|---|---|---|---|---|
| AckedState | HasProperty | Variable | TrueState | LocalizedText | PropertyType | X |
| AckedState | HasProperty | Variable | FalseState | LocalizedText | PropertyType | X |
| ConfirmedState | HasProperty | Variable | TrueState | LocalizedText | PropertyType | X |
| ConfirmedState | HasProperty | Variable | FalseState | LocalizedText | PropertyType | X |

After Table 33 - AlarmConditionType table

**Table 33.a – AlarmConditionType Additional Subcomponents**

| BrowsePath | References | NodeClass | BrowseName | DataType | TypeDefinition | Others |
|---|---|---|---|---|---|---|
| ActiveState | HasProperty | Variable | TrueState | LocalizedText | PropertyType | X |
| ActiveState | HasProperty | Variable | FalseState | LocalizedText | PropertyType | X |
| SuppressedState | HasProperty | Variable | TrueState | LocalizedText | PropertyType | X |
| SuppressedState | HasProperty | Variable | FalseState | LocalizedText | PropertyType | X |
| OutOfServiceState | HasProperty | Variable | TrueState | LocalizedText | PropertyType | X |
| OutOfServiceState | HasProperty | Variable | FalseState | LocalizedText | PropertyType | X |
| InputNodeState | HasProperty | Variable | TrueState | LocalizedText | PropertyType | X |
| InputNodeState | HasProperty | Variable | FalseState | LocalizedText | PropertyType | X |
| LatchedState | HasProperty | Variable | TrueState | LocalizedText | PropertyType | X |
| LatchedState | HasProperty | Variable | FalseState | LocalizedText | PropertyType | X |

After Table 47 - NonExclusiveLimitAlarmType definition table

**Table 47.a – NonExclusiveLimitAlarmType Additional Subcomponents**

| BrowsePath | References | NodeClass | BrowseName | DataType | TypeDefinition | Others |
|---|---|---|---|---|---|---|

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| HighHighState | HasProperty | Variable | TrueState | LocalizedText | PropertyType | X | |
| HighHighState | HasProperty | Variable | FalseState | LocalizedText | PropertyType | X | |
| HighState | HasProperty | Variable | TrueState | LocalizedText | PropertyType | X | |
| HighState | HasProperty | Variable | FalseState | LocalizedText | PropertyType | X | |
| LowState | HasProperty | Variable | TrueState | LocalizedText | PropertyType | X | |
| LowState | HasProperty | Variable | FalseState | LocalizedText | PropertyType | X | |
| LowLowState | HasProperty | Variable | TrueState | LocalizedText | PropertyType | X | |
| LowLowState | HasProperty | Variable | FalseState | LocalizedText | PropertyType | X | |

## OPC UA Specification: Part 10 – Programs

| | |
|---|---|
| **Topic** | Inconsistent ProgramState numbers |
| **Errata Version** | 1.03.4 |
| **Spec Reference** | Part 10, Table 10 “Program States” |
| **Mantis Reference** | 0003927 |
| **Problem Statement** | ProgramState numbers are inconsistent between the Part 10 specification and the UA NodeSet. |
| **Solution** | Since implementations could be based on either the specification or the NodeSet, there is no guarantee for interoperability.<br>Therefore, the following new state numbers are assigned and will be used in both specification and NodeSet starting with version 1.04:<br>• Halted = 11 (was originally 1)<br>• Ready = 12 (was originally 2)<br>• Running = 13 (was originally 3)<br>• Suspended = 14 (was originally 4) |

## OPC UA Specification: Part 11 – Historical Access

| | |
|---|---|
| **Topic** | How to Insert the first annotation? |
| **Errata Version** | 1.03.4 |
| **Spec Reference** | Part 11<br>5.1.2 Annotations Property |
| **Mantis Reference** | 0002558 |
| **Problem Statement** | There is no description on how this property is used or when it exists. |
| **Solution** | Original text for 5.1.2<br>The *DataVariable* or *Object* that has *Annotation* data will add the *Annotations Property* as shown in Table . |

**Table 2 – Annotations Property**

| Name | Use | Data Type | Description |
|---|---|---|---|
| Standard Properties | | | |
| Annotations | O | Annotation | The *Annotations Property* is used to indicate that *Annotation* data exists for the history collection exposed by a *HistricalDataNode*. *Annotation DataType* is defined in Subclause 5.5. |

Since it is not allowed for *Properties* to have *Properties*, the *Annotation Property* is only available for *DataVariables* or *Objects*.

Not every *HistoricalDataNode* in the *AddressSpace* might contain *Annotation* data. The *Annotations Property* indicates whether or not a *HistoricalDataNode* supports *Annotations*. *Annotation* data is accessed using the standard *HistoryRead* functions. *Annotations* are modified, inserted or deleted using the standard *HistoryUpdate* functions.

As with all *HistoricalNodes*, modifications, deletions or additions of *Annotations* will raise the appropriate Historical Audit Event with the corresponding *NodeId*.

New text for 5.1.2

The *DataVariable* or *Object* that has *Annotation* data will add the *Annotations Property* as shown in Table .

**Table 1 – Annotations Property**

| Name | Use | Data Type | Description |
|---|---|---|---|
| Standard Properties | | | |
| Annotations | O | Annotation | The *Annotations Property* is used to indicate that the history collection exposed by a *HistoricalDataNode supports Annotation* data. *Annotation DataType* is defined in Subclause 5.5. |

Since it is not allowed for *Properties* to have *Properties*, the *Annotation Property* is only available for *DataVariables* or *Objects*.

The *Annotations Property* shall be present on every *HistoricalDataNode that supports* modifications, deletions or additions of *Annotations*. Not every *HistoricalDataNode* in the *AddressSpace* might support *Annotation* data. *Annotation* data is accessed using the standard *HistoryRead* functions. *Annotations* are modified, inserted or deleted using the standard *HistoryUpdate* functions. The presence of the *Annotations Property* does not indicate the presence of *Annotations* on the *HistoricalDataNode.*

A *Server* shall allow adding the *Annotation Property* on an existing *HistoricalDataNode* only if it will also support *Annotations* on that *HistoricalDataNode*. A *Server* shall remove any *Annotation* data if it removes the *Annotation Property* from an existing *HistoricalDataNode*.

As with all *HistoricalNodes*, modifications, deletions or additions of *Annotations* will raise the appropriate Historical Audit Event with the corresponding *NodeId*.

<table>
<tr><td>Topic</td><td>There is mismatch in modificationInfo member order in specification and Types.xsd.</td></tr>
<tr><td>Errata Version</td><td>1.03.4</td></tr>
<tr><td>Spec Reference</td><td>Part 11<br>6.5.3 Table 24 HistoryModifiedData Details</td></tr>
<tr><td>Mantis Reference</td><td>0003518</td></tr>
<tr><td>Problem Statement</td><td>The mismatch between argument order has been fixed with the order in the Types.xsd being used.</td></tr>
<tr><td>Solution</td><td>Original table
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>dataValues[]</td><td>DataValue</td><td>An array of values of history data for the Node. The size of the array depends on the requested data parameters.</td></tr>
<tr><td>modificationInfos[]</td><td>ModificationInfo</td><td></td></tr>
<tr><td>Username</td><td>String</td><td>The name of the user that made the modification. Support for this field is optional. A null shall be returned if it is not defined.</td></tr>
<tr><td>modificationTime</td><td>UtcTime</td><td>The time the modification was made. Support for this field is optional. A null shall be returned if it is not defined.</td></tr>
<tr><td>updateType</td><td>HistoryUpdateType</td><td>The modification type for the item.</td></tr>
</table>
New table
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>dataValues[]</td><td>DataValue</td><td>An array of values of history data for the Node. The size of the array depends on the requested data parameters.</td></tr>
<tr><td>modificationInfos[]</td><td>ModificationInfo</td><td></td></tr>
<tr><td>modificationTime</td><td>UtcTime</td><td>The time the modification was made. Support for this field is optional. A null shall be returned if it is not defined.</td></tr>
<tr><td>updateType</td><td>HistoryUpdateType</td><td>The modification type for the item.</td></tr>
<tr><td>Username</td><td>String</td><td>The name of the user that made the modification. Support for this field is optional. A null shall be returned if it is not defined.</td></tr>
</table></td></tr>
</table>

| | |
|---|---|
| **Topic** | Need a new Event sub Audit type for adding Annotations |
| **Errata Version** | 1.03.4 |
| **Spec Reference** | Part 11<br>5.6.4 AuditHistoryStructuredDeataUpdateEventType |
| **Mantis Reference** | 0003876 |
| **Problem Statement** | Audit events for Structured data were missing from part 11. |

**Solution**

The new event AuditHistoryStructuredDataUpdateEventType was added for when working with Structured Data.

Added sub clause 5.6.4:

5.6.4 AuditHistoryStructuredDataUpdateEventType

This is a subtype of *AuditHistoryUpdateEventType* and is used for categorization of structured data update related *Events*. This type follows all the behaviour of its parent type. Its representation in the *AddressSpace* is formally defined in Table 2.

**Table 2 – AuditHistoryStructuredDataUpdateEventType definition**

| **Attribute** | **Value** | | | | |
|---|---|---|---|---|---|
| BrowseName | AuditHistoryStructuredDataUpdateEventType | | | | |
| IsAbstract | False | | | | |
| **References** | **NodeClass** | **BrowseName** | **DataType** | **TypeDefinition** | **ModellingRule** |
| Subtype of the *AuditHistoryUpdateEventType* defined in Part 3, i.e. it has *HasProperty References* to the same *Nodes*. | | | | | |
| HasProperty | Variable | UpdatedNode | NodeId | PropertyType | Mandatory |
| HasProperty | Variable | PerformInsertReplace | PerformUpdateType | PropertyType | Mandatory |
| HasProperty | Variable | NewValues | DataValue[] | PropertyType | Mandatory |
| HasProperty | Variable | OldValues | DataValue[] | PropertyType | Mandatory |

This *EventType* inherits all *Properties* of the *AuditHistoryUpdateEventType*. Their semantic is defined in Part 5.

The *UpdatedNode* identifies the *Attribute* that was written on the *SourceNode*.

The *PerformInsertReplace* enumeration reflects the parameter on the *Service* call.

The *NewValues* identify the value that was written to the *Event*. In the case of a remove it is expected to be a null value.

The *OldValues* identify the value that the *Event* contained before the write. It is acceptable for a *Server* that does not have this information to report a null value. In the case of an insert or remove it is expected to be a null value.

Both the *NewValues* and the *OldValues* will contain a value in the *DataType* and encoding used for writing the value.

| | |
|---|---|
| **Topic** | Determining the first historical data point |
| **Errata Version** | 1.03.4 |
| **Spec Reference** | Part 11<br>Annex A.2 |
| **Mantis Reference** | 0003632 |
| **Problem Statement** | There is a more intuitive way to get the first available point for a historical node. |
| **Solution** | Appended text:<br><br>A third mechanism that can be used is the following:<br>returnBounds=false<br>numValuesPerNode=1<br>startTime=DateTime.MinValue<br>endTime= DateTime.MaxValue |

## OPC UA Specification: Part 13 – Aggregates

| | |
|---|---|
| **Topic** | Variance and SD aggregates have Simple bounds listed when it should be None |
| **Errata Version** | 1.03.4 |
| **Spec Reference** | Part 13 Aggregates |
| **Mantis Reference** | 0003300<br><br>0003312 |
| **Problem Statement** | The specification was incorrect in listing use Simple bounding values for the Stadard Deviation and Variance aggregates. The tables for these aggregates (Table 48, 49, 50, and 51) have been changed to not use bounding values. |

**Solution**

Original Table 48 StandardDeviationSample Aggregate summary

| **StandardDeviationSample Aggregate Characteristics** | |
|---|---|
| Type | Calculated |
| Data Type | Status Code |
| Use Bounds | Simple |
| Timestamp | StartTime |
| | |
| **Status Code Calculations** | |
| Calculation Method | Custom<br>Always Good |
| Partial | Set Sometimes<br>If an interval is not a complete interval |
| Calculated | Set Always |
| Interpolated | Not Set |
| Raw | Not Set |
| Multi Value | Not Set |
| | |
| **Status Code Common Special Cases** | |
| Before Start of Data | Bad_NoData |
| After End of Data | Bad_NoData |
| No Start Bound | No special handing required |
| No End Bound | No special handing required |
| Bound Bad | No special handing required |
| Bound Uncertain | No special handing required |

Revised table

| **StandardDeviationSample Aggregate Characteristics** | |
|---|---|
| Type | Calculated |
| Data Type | Status Code |
| Use Bounds | None |
| Timestamp | StartTime |
| | |
| **Status Code Calculations** | |
| Calculation Method | Custom<br>Always Good |
| Partial | Set Sometimes<br>If an interval is not a complete interval |
| Calculated | Set Always |
| Interpolated | Not Set |
| Raw | Not Set |
| Multi Value | Not Set |
| | |
| **Status Code Common Special Cases** | |
| Before Start of Data | Bad_NoData |
| After End of Data | Bad_NoData |
| No Start Bound | No special handing required |
| No End Bound | No special handing required |
| Bound Bad | No special handing required |
| Bound Uncertain | No special handing required |

Tables 49, 50, and 51 have had a similar change.

| | |
|---|---|
| **Topic** | StandardDeviation and Variance aggregate examples for Historian1 using an uncertain value when it should not |
| **Errata Version** | 1.03.4 |
| **Spec Reference** | Part 13 Aggregates<br>Annex A.35.2 StandardDeviationSample data<br>Annex A.36.2 VarianceSample data<br>Annex A.37.2 StandardDeviationPopulation data<br>Annex A.38.2 VariancePopulation data |
| **Mantis Reference** | 0003301 0003313 |
| **Problem Statement** | Historian 1 examples for StandardDeviationSample, VarianceSample, StandardDeviationPopulation, and VariancePopulation used Non-Good data in the calculation. They should use only Good data. |

**Solution**

Original tables

Annex A.35.2 StandardDeviationSample data

| Historian1 | | | |
|---|---|---|---|
| **Timestamp** | **Value** | **StatusCode** | **Notes** |
| 12:00:00.000 | 0 | Good, Calculated, Partial | |
| 12:00:20.000 | 7.071 | Good, Calculated | |
| 12:00:40.000 | 0 | UncertainDataSubNormal, Calculated | |
| 12:01:00.000 | 7.071 | UncertainDataSubNormal, Calculated | |
| 12:01:20.000 | 7.071 | Good, Calculated, Partial | |

Annex A.36.2 VarianceSample data

| Historian1 | | | |
|---|---|---|---|
| **Timestamp** | **Value** | **StatusCode** | **Notes** |
| 12:00:00.000 | 0 | Good, Calculated, Partial | |
| 12:00:20.000 | 50 | Good, Calculated | |
| 12:00:40.000 | 0 | UncertainDataSubNormal, Calculated | |
| 12:01:00.000 | 50 | UncertainDataSubNormal, Calculated | |
| 12:01:20.000 | 50 | Good, Calculated, Partial | |

Annex A.37.2 StandardDeviationPopulation data

| Historian1 | | | |
|---|---|---|---|
| **Timestamp** | **Value** | **StatusCode** | **Notes** |
| 12:00:00.000 | 0 | Good, Calculated, Partial | |
| 12:00:20.000 | 5 | Good, Calculated | |
| 12:00:40.000 | 0 | UncertainDataSubNormal, Calculated | |
| 12:01:00.000 | 5 | UncertainDataSubNormal, Calculated | |
| 12:01:20.000 | 5 | Good, Calculated, Partial | |

Annex A.38.2 VariancePopulation data

| Historian1 | | | |
|---|---|---|---|
| **Timestamp** | **Value** | **StatusCode** | **Notes** |
| 12:00:00.000 | 0 | Good, Calculated, Partial | |
| 12:00:20.000 | 25 | Good, Calculated | |
| 12:00:40.000 | 0 | UncertainDataSubNormal, Calculated | |
| 12:01:00.000 | 25 | UncertainDataSubNormal, Calculated | |
| 12:01:20.000 | 25 | Good, Calculated, Partial | |

Revised tables

Annex A.35.2 StandardDeviationSample data

| Historian1 | | | |
|---|---|---|---|
| **Timestamp** | **Value** | **StatusCode** | **Notes** |
| 12:00:00.000 | 0 | Good, Calculated, Partial | |
| 12:00:20.000 | 7.071 | Good, Calculated | |
| 12:00:40.000 | 0 | UncertainDataSubNormal, Calculated | |
| 12:01:00.000 | 0 | UncertainDataSubNormal, Calculated | |
| 12:01:20.000 | 7.071 | Good, Calculated, Partial | |

Annex A.36.2 VarianceSample data

| Historian1 | | | |
|---|---|---|---|
| **Timestamp** | **Value** | **StatusCode** | **Notes** |
| 12:00:00.000 | 0 | Good, Calculated, Partial | |
| 12:00:20.000 | 50 | Good, Calculated | |

| | | | |
|---|---|---|---|
| 12:00:40.000 | 0 | UncertainDataSubNormal, Calculated | |
| 12:01:00.000 | 0 | UncertainDataSubNormal, Calculated | |
| 12:01:20.000 | 50 | Good, Calculated, Partial | |

Annex A.37.2 StandardDeviationPopulation data

| Historian1 | | | |
|---|---|---|---|
| **Timestamp** | **Value** | **StatusCode** | **Notes** |
| 12:00:00.000 | 0 | Good, Calculated, Partial | |
| 12:00:20.000 | 5 | Good, Calculated | |
| 12:00:40.000 | 0 | UncertainDataSubNormal, Calculated | |
| 12:01:00.000 | 0 | UncertainDataSubNormal, Calculated | |
| 12:01:20.000 | 5 | Good, Calculated, Partial | |

Annex A.38.2 VariancePopulation data

| Historian1 | | | |
|---|---|---|---|
| **Timestamp** | **Value** | **StatusCode** | **Notes** |
| 12:00:00.000 | 0 | Good, Calculated, Partial | |
| 12:00:20.000 | 25 | Good, Calculated | |
| 12:00:40.000 | 0 | UncertainDataSubNormal, Calculated | |
| 12:01:00.000 | 0 | UncertainDataSubNormal, Calculated | |
| 12:01:20.000 | 25 | Good, Calculated, Partial | |

<table>
<tr><td>Topic</td><td>Delta Aggregate example for Historian1 has UncertainDataSubnormal value instead of BadNoData</td></tr>
<tr><td>Errata Version</td><td>1.03.4</td></tr>
<tr><td>Spec Reference</td><td>Part 13 Aggregates<br>Annex A.27.2 Delta</td></tr>
<tr><td>Mantis Reference</td><td>0003302</td></tr>
<tr><td>Problem Statement</td><td>The interval starting at 12:01:04.000 has UncertainDataSubnormal quality instead of BadNoData.</td></tr>
<tr><td>Solution</td><td>Original table
<table>
<tr><th colspan="4">Historian1</th></tr>
<tr><th>Timestamp</th><th>Value</th><th>StatusCode</th><th>Notes</th></tr>
<tr><td>12:00:00.000</td><td>0</td><td>Good, Calculated, Partial</td><td></td></tr>
<tr><td>12:00:16.000</td><td>10</td><td>Good, Calculated</td><td></td></tr>
<tr><td>12:00:32.000</td><td>0</td><td>BadNoData</td><td></td></tr>
<tr><td>12:00:48.000</td><td>10</td><td>Good, Calculated</td><td></td></tr>
<tr><td>12:01:04.000</td><td>0</td><td>UncertainDataSubnormal, Calculated</td><td></td></tr>
<tr><td>12:01:20.000</td><td>10</td><td>Good, Calculated, Partial</td><td></td></tr>
<tr><td>12:01:36.000</td><td></td><td>BadNoData</td><td></td></tr>
</table>
Revised table
<table>
<tr><th colspan="4">Historian1</th></tr>
<tr><th>Timestamp</th><th>Value</th><th>StatusCode</th><th>Notes</th></tr>
<tr><td>12:00:00.000</td><td>0</td><td>Good, Calculated, Partial</td><td></td></tr>
<tr><td>12:00:16.000</td><td>10</td><td>Good, Calculated</td><td></td></tr>
<tr><td>12:00:32.000</td><td>0</td><td>BadNoData</td><td></td></tr>
<tr><td>12:00:48.000</td><td>10</td><td>Good, Calculated</td><td></td></tr>
<tr><td>12:01:04.000</td><td>0</td><td>BadNoData</td><td></td></tr>
<tr><td>12:01:20.000</td><td>10</td><td>Good, Calculated, Partial</td><td></td></tr>
<tr><td>12:01:36.000</td><td></td><td>BadNoData</td><td></td></tr>
</table></td></tr>
</table>

| | |
|---|---|
| **Topic** | Interpolative example for Historian2 has the wrong value in the last interval |
| **Errata Version** | 1.03.4 |
| **Spec Reference** | Part 13 Aggregates<br>Annex A.2.2 |
| **Mantis Reference** | 0003310 |
| **Problem Statement** | The last interval for Interpolative Historian 2 has the value (102.500) as if the *UseSlopedExtrapolation* parameter is true but the example specifically has it set to false. |

**Solution**

The original table for Historian 2 with the incorrect last value for the last interval.

| Historian2 | | | |
|---|---|---|---|
| **Timestamp** | **Value** | **StatusCode** | **Notes** |
| 12:00:00.000 | | BadNoData | |
| 12:00:05.000 | 11.304 | Good, Interpolated | |
| 12:00:10.000 | 13.478 | Good, Interpolated | |
| 12:00:15.000 | 15.652 | Good, Interpolated | |
| 12:00:20.000 | 17.826 | Good, Interpolated | |
| 12:00:25.000 | 20 | Good | |
| 12:00:30.000 | 25.909 | Good, Interpolated | |
| 12:00:35.000 | 28.182 | Good, Interpolated | |
| 12:00:40.000 | 31.111 | UncertainDataSubNormal, Interpolated | |
| 12:00:45.000 | 36.667 | UncertainDataSubNormal, Interpolated | |
| 12:00:50.000 | 45 | Good, Interpolated | |
| 12:00:55.000 | 51.500 | Good, Interpolated | |
| 12:01:00.000 | 54 | Good, Interpolated | |
| 12:01:05.000 | 56.500 | Good, Interpolated | |
| 12:01:10.000 | 59 | Good, Interpolated | |
| 12:01:15.000 | 62.727 | UncertainDataSubNormal, Interpolated | |
| 12:01:20.000 | 67.273 | UncertainDataSubNormal, Interpolated | |
| 12:01:25.000 | 76.667 | Good, Interpolated | |
| 12:01:30.000 | 90 | Good | |
| 12:01:35.000 | 102.500 | UncertainDataSubNormal, Interpolated | |

The corrected table for historian 2.

| Historian2 | | | |
|---|---|---|---|
| **Timestamp** | **Value** | **StatusCode** | **Notes** |
| 12:00:00.000 | | BadNoData | |
| 12:00:05.000 | 11.304 | Good, Interpolated | |
| 12:00:10.000 | 13.478 | Good, Interpolated | |
| 12:00:15.000 | 15.652 | Good, Interpolated | |
| 12:00:20.000 | 17.826 | Good, Interpolated | |
| 12:00:25.000 | 20 | Good | |
| 12:00:30.000 | 25.909 | Good, Interpolated | |
| 12:00:35.000 | 28.182 | Good, Interpolated | |
| 12:00:40.000 | 31.111 | UncertainDataSubNormal, Interpolated | |
| 12:00:45.000 | 36.667 | UncertainDataSubNormal, Interpolated | |
| 12:00:50.000 | 45 | Good, Interpolated | |
| 12:00:55.000 | 51.500 | Good, Interpolated | |
| 12:01:00.000 | 54 | Good, Interpolated | |
| 12:01:05.000 | 56.500 | Good, Interpolated | |
| 12:01:10.000 | 59 | Good, Interpolated | |
| 12:01:15.000 | 62.727 | UncertainDataSubNormal, Interpolated | |
| 12:01:20.000 | 67.273 | UncertainDataSubNormal, Interpolated | |
| 12:01:25.000 | 76.667 | Good, Interpolated | |
| 12:01:30.000 | 90 | Good | |
| 12:01:35.000 | 90 | UncertainDataSubNormal, Interpolated | |

| | |
|---|---|
| **Topic** | Contradiction in Timestamp definition of at least Minimum and Maximum Aggregates |
| **Errata Version** | 1.03.4 |
| **Spec Reference** | Part 13 Aggregates<br>5.4.3.10 Minimum<br>5.4.3.11 Maximum |
| **Mantis Reference** | 0003391 |
| **Problem Statement** | There is a contradiction in what the interval timestamp should be. |
| **Solution** | 5.4.3.10 Minimum<br>The original text in paragraph one was<br>"The Minimum *Aggregate* defined in Table 21 retrieves the minimum Good raw value within the interval, and returns that value with the timestamp at which that value occurs. Note that if the same minimum exists at more than one timestamp, the oldest one is retrieved and the *MultipleValues* bit is set*."*<br><br>It has been changed to<br><br>*"*The Minimum *Aggregate* defined in Table 21 retrieves the minimum Good raw value within the interval, and returns that value with the timestamp at the start of the interval. Note that if the same minimum exists at more than one timestamp, the oldest one is retrieved and the *MultipleValues* bit is set."<br><br>5.4.3.11 Maximum<br>The original text in paragraph one was<br>"The Maximum *Aggregate* defined in Table 22 retrieves the maximum Good raw value within the interval, and returns that value with the timestamp at which that value occurs. Note that if the same maximum exists at more than one timestamp, the oldest one is retrieved and the *MultipleValues* bit is set*."*<br><br>It has been changed to<br><br>*"*The Maximum *Aggregate* defined in Table 22 retrieves the maximum Good raw value within the interval, and returns that value with the timestamp at the start of the interval. Note that if the same maximum exists at more than one timestamp, the oldest one is retrieved and the *MultipleValues* bit is set." |

| | |
|---|---|
| **Topic** | MinimumActualTime2 has the wrong Aggregate for the base aggregate calculation |
| **Errata Version** | 1.03.4 |
| **Spec Reference** | Part 13 Aggregates<br>5.4.3.17 MinimumActualTime2 |
| **Mantis Reference** | 0003589 |
| **Problem Statement** | MinimumActualTime2 says it uses the Minimum aggregate as its example but it actually should be using the MinimumActualTime aggregate. |
| **Solution** | 5.4.3.17 MinimumActualTime2<br>The first sentence original text in paragraph one was<br>"The MinimumActualTime2 *Aggregate* defined in Table 28 retrieves the minimum Good value for each interval as defined for Minimum except that *Simple Bounding Values* are included*."*<br>It has been changed to<br>*"*The MinimumActualTime2 *Aggregate* defined in Table 28 retrieves the minimum Good value for each interval as defined for MinimumActualTime except that *Simple Bounding Values* are included." |

| | |
|---|---|
| **Topic** | Delta Aggregate should always have Calculated flag set |
| **Errata Version** | 1.03.4 |
| **Spec Reference** | Part 13 Aggregates<br>Table 32 Delta Aggregate Summary |
| **Mantis Reference** | 0003590 |
| **Problem Statement** | The Delta aggregate should always have the calculated bit set and is wrong in the aggregate table 38. |

**Solution**

Original table

| **Delta Aggregate Characteristics** | |
|---|---|
| Type | Calculated |
| Data Type | Same as Source |
| Use Bounds | None |
| Timestamp | StartTime |
| | |
| **Status Code Calculations** | |
| Calculation Method | Custom<br>*Uncertain_DataSubNormal* if non-Good values are skipped while looking for the first or last values |
| Partial | Set Sometimes<br>If an interval is not a complete interval |
| Calculated | Not Set |
| Interpolated | Not Set |
| Raw | Always |
| Multi Value | Not Set |
| | |
| **Status Code Common Special Cases** | |
| Before Start of Data | Bad_NoData |
| After End of Data | Bad_NoData |
| No Start Bound | Does not apply |
| No End Bound | Does not apply |
| Bound Bad | Does not apply |
| Bound Uncertain | Does not apply |

Revised table

| **Delta Aggregate Characteristics** | |
|---|---|
| Type | Calculated |
| Data Type | Same as Source |
| Use Bounds | None |
| Timestamp | StartTime |
| | |
| **Status Code Calculations** | |
| Calculation Method | Custom<br>*Uncertain_DataSubNormal* if non-Good values are skipped while looking for the first or last values |
| Partial | Set Sometimes<br>If an interval is not a complete interval |
| Calculated | Set Always |
| Interpolated | Not Set |
| Raw | Always |
| Multi Value | Not Set |
| | |
| **Status Code Common Special Cases** | |
| Before Start of Data | Bad_NoData |
| After End of Data | Bad_NoData |
| No Start Bound | Does not apply |
| No End Bound | Does not apply |
| Bound Bad | Does not apply |
| Bound Uncertain | Does not apply |

<table>
<tr><td>Topic</td><td>Aggregate StandardDeviationPopulation Historian2 has an error in one interval.</td></tr>
<tr><td>Errata Version</td><td>1.03.4</td></tr>
<tr><td>Spec Reference</td><td>Part 13 Aggregates<br>Annex A.37.2 StandardDeviationPopulation data</td></tr>
<tr><td>Mantis Reference</td><td>0003595</td></tr>
<tr><td>Problem Statement</td><td>Aggregate StandardDeviationPopulation Historian2 has wrong value for interval starting at 12:00:40.000.</td></tr>
<tr><td>Solution</td><td>Original table
<table>
<tr><th colspan="4">Historian2</th></tr>
<tr><th>Timestamp</th><th>Value</th><th>StatusCode</th><th>Notes</th></tr>
<tr><td>12:00:00.000</td><td>0</td><td>Good, Calculated, Partial</td><td></td></tr>
<tr><td>12:00:20.000</td><td>4.082</td><td>Good, Calculated</td><td></td></tr>
<tr><td>12:00:40.000</td><td>4</td><td>UncertainDataSubNormal, Calculated</td><td></td></tr>
<tr><td>12:01:00.000</td><td>0</td><td>UncertainDataSubNormal, Calculated</td><td></td></tr>
<tr><td>12:01:20.000</td><td>8.165</td><td>Good, Calculated, Partial</td><td></td></tr>
</table>
Revised table
<table>
<tr><th colspan="4">Historian2</th></tr>
<tr><th>Timestamp</th><th>Value</th><th>StatusCode</th><th>Notes</th></tr>
<tr><td>12:00:00.000</td><td>0</td><td>Good, Calculated, Partial</td><td></td></tr>
<tr><td>12:00:20.000</td><td>4.082</td><td>Good, Calculated</td><td></td></tr>
<tr><td>12:00:40.000</td><td>5</td><td>UncertainDataSubNormal, Calculated</td><td></td></tr>
<tr><td>12:01:00.000</td><td>0</td><td>UncertainDataSubNormal, Calculated</td><td></td></tr>
<tr><td>12:01:20.000</td><td>8.165</td><td>Good, Calculated, Partial</td><td></td></tr>
</table></td></tr>
</table>

| | |
|---|---|
| **Topic** | TimeAverage Aggregate is using stepped calculation for Historian3 and it shouldn't. |
| **Errata Version** | 1.03.4 |
| **Spec Reference** | Part 13 Aggregates<br>Annex A.4.2 TimeAverage |
| **Mantis Reference** | 0003604 |
| **Problem Statement** | The TimeAverage aggregate explicitly states that the stepped setting isn't used and all calculations will use sloped. The table in A.4.2 Historian3 has the values as if stepped is used. |

**Solution**

Original table

| Historian3 | | | |
|---|---|---|---|
| **Timestamp** | **Value** | **StatusCode** | **Notes** |
| 12:00:00.000 | 10 | UncertainDataSubNormal, Calculated, Partial | |
| 12:00:05.000 | 10 | Good, Calculated | |
| 12:00:10.000 | 10 | Good, Calculated | |
| 12:00:15.000 | 10 | Good, Calculated | |
| 12:00:20.000 | 10 | Good, Calculated | |
| 12:00:25.000 | 22 | Good, Calculated | |
| 12:00:30.000 | 25 | Good, Calculated | |
| 12:00:35.000 | 26 | Good, Calculated | |
| 12:00:40.000 | 30 | UncertainDataSubNormal, Calculated | |
| 12:00:45.000 | 34 | UncertainDataSubNormal, Calculated | |
| 12:00:50.000 | 46 | Good, Calculated | |
| 12:00:55.000 | 50 | Good, Calculated | |
| 12:01:00.000 | 50 | Good, Calculated | |
| 12:01:05.000 | 50 | Good, Calculated | |
| 12:01:10.000 | 56 | Good, Calculated | |
| 12:01:15.000 | 60 | UncertainDataSubNormal, Calculated | |
| 12:01:20.000 | 64 | UncertainDataSubNormal, Calculated | |
| 12:01:25.000 | 78 | Good, Calculated | |
| 12:01:30.000 | 90 | UncertainDataSubNormal, Calculated | |
| 12:01:35.000 | 90 | UncertainDataSubNormal, Calculated | |

Revised table

| Historian3 | | | |
|---|---|---|---|
| **Timestamp** | **Value** | **StatusCode** | **Notes** |
| 12:00:00.000 | 10.652 | UncertainDataSubNormal, Calculated, Partial | |
| 12:00:05.000 | 12.391 | Good, Calculated | |
| 12:00:10.000 | 14.565 | Good, Calculated | |
| 12:00:15.000 | 16.739 | Good, Calculated | |
| 12:00:20.000 | 18.913 | Good, Calculated | |
| 12:00:25.000 | 23.682 | Good, Calculated | |
| 12:00:30.000 | 27.046 | Good, Calculated | |
| 12:00:35.000 | 29.384 | UncertainDataSubNormal, Calculated | |
| 12:00:40.000 | 33.889 | UncertainDataSubNormal, Calculated | |
| 12:00:45.000 | 40 | UncertainDataSubNormal, Calculated | |
| 12:00:50.000 | 49.450 | Good, Calculated | |
| 12:00:55.000 | 52.750 | Good, Calculated | |
| 12:01:00.000 | 55.250 | Good, Calculated | |
| 12:01:05.000 | 57.750 | Good, Calculated | |
| 12:01:10.000 | 60.618 | UncertainDataSubNormal, Calculated | |
| 12:01:15.000 | 65 | UncertainDataSubNormal, Calculated | |

| | | | | |
|---|---|---|---|---|
| | 12:01:20.000 | 70.515 | UncertainDataSubNormal, Calculated | |
| | 12:01:25.000 | 83.667 | Good, Calculated | |
| | 12:01:30.000 | 90 | UncertainDataSubNormal, Calculated, Partial | |
| | 12:01:35.000 | 90 | UncertainDataSubNormal, Calculated | |

| | |
|---|---|
| **Topic** | The Aggregate Number of Transitions has some errors. |
| **Errata Version** | 1.03.4 |
| **Spec Reference** | Part 13 Aggregates<br>Annex A.37 NumberOfTransitions |
| **Mantis Reference** | 0003605 |
| **Problem Statement** | The heading in the A.22.1 has an interval of 00:00:05.000 but the tables have an interval of 00:00:16.000.<br><br>The table for Historian1 has the wrong value for the 12:01:20.000 interval. It should be 2 not 1.<br><br>The first interval for all Historian examples needs to be 1. The first good value at the start of the request is always counted as a transition. |

**Solution**

Original text for description

"The following examples demonstrate NumberOfTransitions *Aggregate* scenarios. ***ProcessingInterval***: 00:00:05, ***StartTime***: 12:00:00, ***EndTime***: 12:01:40."

Is now

"The following examples demonstrate NumberOfTransitions *Aggregate* scenarios. ***ProcessingInterval***: 00:00:16, ***StartTime***: 12:00:00, ***EndTime***: 12:01:40."

Original tables

| **Historian1** | | | |
|---|---|---|---|
| **Timestamp** | **Value** | **StatusCode** | **Notes** |
| 12:00:00.000 | 0 | Good, Calculated, Partial | |
| 12:00:16.000 | 2 | Good, Calculated | |
| 12:00:32.000 | | Bad | |
| 12:00:48.000 | 2 | Good, Calculated | |
| 12:01:04.000 | 1 | UncertainDataSubNormal, Calculated | |
| 12:01:20.000 | 1 | Good, Calculated, Partial | |
| 12:01:36.000 | | BadNoData | |

| **Historian2** | | | |
|---|---|---|---|
| **Timestamp** | **Value** | **StatusCode** | **Notes** |
| 12:00:00.000 | 0 | Good, Calculated, Partial | |
| 12:00:16.000 | 2 | Good, Calculated | |
| 12:00:32.000 | 1 | UncertainDataSubNormal, Calculated | |
| 12:00:48.000 | 2 | Good, Calculated | |
| 12:01:04.000 | 1 | UncertainDataSubNormal, Calculated | |
| 12:01:20.000 | 3 | Good, Calculated, Partial | |
| 12:01:36.000 | | BadNoData | |

| **Historian3** | | | |
|---|---|---|---|
| **Timestamp** | **Value** | **StatusCode** | **Notes** |
| 12:00:00.000 | 0 | Good, Calculated, Partial | |
| 12:00:16.000 | 2 | Good, Calculated | |
| 12:00:32.000 | | Bad | |
| 12:00:48.000 | 2 | Good, Calculated | |
| 12:01:04.000 | 1 | Good, Calculated | |
| 12:01:20.000 | 3 | Good, Calculated, Partial | |
| 12:01:36.000 | | BadNoData | |

| **Historian4** | | | |
|---|---|---|---|
| **Timestamp** | **Value** | **StatusCode** | **Notes** |
| 12:00:00.000 | 0 | Good, Calculated, Partial | |
| 12:00:16.000 | 2 | Good, Calculated | |
| 12:00:32.000 | 0 | UncertainDataSubNormal, Calculated | |
| 12:00:48.000 | 1 | Good, Calculated | |
| 12:01:04.000 | 0 | UncertainDataSubNormal, Calculated | |

| | | | |
|---|---|---|---|
| 12:01:20.000 | 3 | Good, Calculated, Partial | |
| 12:01:36.000 | | BadNoData | |

Revised tables

| Historian1 | | | |
|---|---|---|---|
| **Timestamp** | **Value** | **StatusCode** | **Notes** |
| 12:00:00.000 | 1 | Good, Calculated, Partial | |
| 12:00:16.000 | 2 | Good, Calculated | |
| 12:00:32.000 | | Bad | |
| 12:00:48.000 | 2 | Good, Calculated | |
| 12:01:04.000 | 1 | UncertainDataSubNormal, Calculated | |
| 12:01:20.000 | 2 | Good, Calculated, Partial | |
| 12:01:36.000 | | BadNoData | |

| Historian2 | | | |
|---|---|---|---|
| **Timestamp** | **Value** | **StatusCode** | **Notes** |
| 12:00:00.000 | 1 | Good, Calculated, Partial | |
| 12:00:16.000 | 2 | Good, Calculated | |
| 12:00:32.000 | 1 | UncertainDataSubNormal, Calculated | |
| 12:00:48.000 | 2 | Good, Calculated | |
| 12:01:04.000 | 1 | UncertainDataSubNormal, Calculated | |
| 12:01:20.000 | 3 | Good, Calculated, Partial | |
| 12:01:36.000 | | BadNoData | |

| Historian3 | | | |
|---|---|---|---|
| **Timestamp** | **Value** | **StatusCode** | **Notes** |
| 12:00:00.000 | 1 | Good, Calculated, Partial | |
| 12:00:16.000 | 2 | Good, Calculated | |
| 12:00:32.000 | | Bad | |
| 12:00:48.000 | 2 | Good, Calculated | |
| 12:01:04.000 | 1 | Good, Calculated | |
| 12:01:20.000 | 3 | Good, Calculated, Partial | |
| 12:01:36.000 | | BadNoData | |

| Historian4 | | | |
|---|---|---|---|
| **Timestamp** | **Value** | **StatusCode** | **Notes** |
| 12:00:00.000 | 1 | Good, Calculated, Partial | |
| 12:00:16.000 | 2 | Good, Calculated | |
| 12:00:32.000 | 0 | UncertainDataSubNormal, Calculated | |
| 12:00:48.000 | 1 | Good, Calculated | |
| 12:01:04.000 | 0 | UncertainDataSubNormal, Calculated | |
| 12:01:20.000 | 3 | Good, Calculated, Partial | |
| 12:01:36.000 | | BadNoData | |

_____