OPC 10000

OPC Unified Architecture

Errata

Release 1.04.12

2024-01-05

OPC UA Specification Errata

Specification Type:	Industry Standard Specification	Comments:	Report or view errata: http://www.opcfoundation.org/errata
Document Number	**OPC 10000**		
Title:	OPC Unified Architecture Errata	Date:	2024-01-05
Version:	Release 1.04.12	Software:	MS-Word
		Source:	OPC 10000 - UA Specification Errata 1.04.12.docx
Author:	OPC Foundation	Status:	Release

CONTENTS

Page

FIGURES iv
TABLES iv
FOREWORD v
AGREEMENT OF USE v
Scope 1
OPC UA Specification: Part 3 – Address Space Model 1
OPC UA Specification: Part 4 – Services 9
OPC UA Specification: Part 5 – Information Model 24
OPC UA Specification: Part 6 – Mappings 38
OPC UA Specification: Part 7 – Profiles 51
OPC UA Specification: Part 8 – DataAccess 52
OPC UA Specification: Part 9 – Alarms & Conditions 55
OPC UA Specification: Part 11 – Historical Access 65
OPC UA Specification: Part 12 – Discovery and Global Services 70
OPC UA Specification: Part 13 – Aggregates 81
OPC UA Specification: Part 14 – PubSub 113
OPC UA Specification: Part 15 – Safety 157
OPC UA Specification: Part 17 – Alias Names 158
OPC UA Specification: Part 19 – Dictionary Reference 161
OPC UA Specification: Part 22 – Base Network Model 162
OPC UA Specification: Amendment 1 – AnalogItem Types 162
OPC UA Specification: Amendment 2 – ChoiceStates and Guards 162
OPC UA Specification: Amendment 3 – Method Metadata 163
OPC UA Specification: Amendment 4 – ECC 164
OPC UA Specification: Amendment 5 – Dictionary Reference 165
OPC UA Specification: Amendment 6 – UADP Header Layouts 165
OPC UA Specification: Amendment 7 – Interfaces and AddIns 166
OPC UA Specification: Amendment 10 – Currency 167
OPC UA Specification: Amendment 11 – Spatial Types 168
OPC UA Specification: Amendment 13 – Ordered List 168
OPC UA Specification: NodeSets and Generated Files 168

FIGURES

No table of figures entries found.

TABLES

Table 1 – SimpleAttributeOperand 10
Table 2 – SimpleAttributeOperand 10
Table 3 – PubSubState state machine 146

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.

Copyright © 2006-2024, OPC Foundation, Inc.

AGREEMENT OF USE

COPYRIGHT RESTRICTIONS

Any unauthorized use of this specification may violate copyright laws, trademark laws, and communications regulations and statutes. This document contains information which is protected by copyright. All Rights Reserved. No part of this work covered by copyright herein may be reproduced or used in any form or by any means--graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems--without permission of the copyright owner.

OPC Foundation members and non-members are prohibited from copying and redistributing this specification. All copies must be obtained on an individual basis, directly from the OPC Foundation Web site http://www.opcfoundation.org.

PATENTS

The attention of adopters is directed to the possibility that compliance with or adoption of OPC specifications may require use of an invention covered by patent rights. OPC shall not be responsible for identifying patents for which a license may be required by any OPC specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. OPC specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

WARRANTY AND LIABILITY DISCLAIMERS

WHILE THIS PUBLICATION IS BELIEVED TO BE ACCURATE, IT IS PROVIDED "AS IS" AND MAY CONTAIN ERRORS OR MISPRINTS. THE OPC FOUDATION MAKES NO WARRANTY OF ANY KIND, EXPRESSED OR IMPLIED, WITH REGARD TO THIS PUBLICATION, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE OR USE. IN NO EVENT SHALL THE OPC FOUNDATION BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, RELIANCE OR COVER DAMAGES, INCLUDING LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED BY ANY USER OR ANY THIRD PARTY IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The entire risk as to the quality and performance of software developed using this specification is borne by you.

RESTRICTED RIGHTS LEGEND

This Specification is provided with Restricted Rights. Use, duplication or disclosure by the U.S. government is subject to restrictions as set forth in (a) this Agreement pursuant to DFARs 227.7202-3(a); (b) subparagraph (c)(1)(i) of the Rights in Technical Data and Computer Software clause at DFARs 252.227-7013; or (c) the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 subdivision (c)(1) and (2), as applicable. Contractor / manufacturer are the OPC Foundation, 16101 N. 82nd Street, Suite 3B, Scottsdale, AZ, 85260-1830

COMPLIANCE

The OPC Foundation shall at all times be the sole entity that may authorize developers, suppliers and sellers of hardware and software to use certification marks, trademarks or other special designations to indicate compliance with these materials. Products developed using this specification may claim compliance or conformance with this specification if and only if the software satisfactorily meets the certification requirements set by the OPC Foundation. Products that do not meet these requirements may claim only that the product was based on this specification and must not claim compliance or conformance with this specification.

TRADEMARKS

Most computer and software brand names have trademarks or registered trademarks. The individual trademarks have not been listed here.

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

Scope

This Errata document contains all the known corrections and additions (via Amendments and new Parts) to OPC UA Specification Parts 1 through 22 for version 1.04. This document is updated regularly when issues are found or enhancements are made between major releases of the Specification.

If anything in the Errata affects the NodeSets, new NodeSets will be published with a reference to the version and date of this Errata document.

OPC UA Specification: Part 3 – Address Space Model

<table>
<tr><td>Topic</td><td>Correction to access restrictions</td></tr>
<tr><td>Errata Version</td><td>1.04.1</td></tr>
<tr><td>Spec Reference</td><td>Part 3
8.56 – AccessRestrictionsType</td></tr>
<tr><td>Mantis Reference</td><td>0004177</td></tr>
<tr><td>Problem Statement</td><td>Currently the AccessRestrictions can only be applied to all permissions which does not allow exclusion of the browse permission. In many cases access to browse a node does not require restrictions but access to a Variables Value may be.</td></tr>
<tr><td>Solution</td><td>The ApplyRestrictionsToBrowse bit has been defined which when not set excludes the Browse Permission from the SigningRequired and EncryptionRequired restrictions.
<table>
<tr><th>Name</th><th>Bit</th><th>Description</th></tr>
<tr><td>SigningRequired</td><td>0</td><td>The Client can only access the Node when using a SecureChannel which digitally signs all messages. This does not apply to the Browse permission if the ApplyRestrictionsToBrowse is not set.</td></tr>
<tr><td>EncryptionRequired</td><td>1</td><td>The Client can only access the Node when using a SecureChannel which encrypts all messages. This does not apply to the Browse permission if the ApplyRestrictionsToBrowse is not set.</td></tr>
<tr><td>SessionRequired</td><td>2</td><td>The Client cannot access the Node when using SessionlessInvoke Service invocation.</td></tr>
<tr><td>ApplyRestrictionsToBrowse</td><td>3</td><td>The access restrictions SigningRequired and EncryptionRequired are also applied to for the Browse permission, if this bit is set.</td></tr>
</table></td></tr>
</table>

Topic	Handling of StructureFields with abstract DataType
Errata Version	1.04.10
Spec Reference	Part 3 8.49 - StructureDefinition 8.51 - StructureField
Mantis Reference	0004421, 0005481
Problem Statement	Part 14 have several structure fields where the DataType is abstract and only concrete subtypes can be used (e.g. TransportSettings and MessageSettings) on the different PubSub object levels. This requires that an ExtensionObject is embedded as the field instead of an embedded structure. While this is correct for the resulting structure when encoded for transport, the concrete abstract type information is lost.

Solution

Replace Table 34 with the following:

Name	Type	Description
StructureDefinition	Structure	
defaultEncodingId	NodeId	The *NodeId* of the default DataTypeEncoding for the *DataType*. The default depends on the message encoding, Default Binary for UA Binary encoding, Default JSON for JSON encoding and Default XML for XML encoding. If the *DataType* is only used inside nested *Structures* and is not directly contained in an *ExtensionObject*, the encoding *NodeId* is null.
baseDataType	NodeId	The *NodeId* of the direct supertype of the *DataType*. This might be the abstract *Structure* or the *Union DataType*.
structureType	StructureType	An enumeration that specifies the type of Structure defined by the *DataType*. The StructureType DataType is defined in 8.49A. Only one of the fields defined for the data type is encoded into a value if the data type is a *Union*.
fields	StructureField []	The list of fields that make up the data type. This definition assumes the structure has a sequential layout. The StructureField *DataType* is defined in 8.51. For *Structures* derived from another *Structure DataType* this list shall begin with the fields of the baseDataType followed by the fields of this *StructureDefinition*.

Add Clause 8.49A as the following:

This *DataType* is an enumeration that specifies type of *Structure* defined by a *StructureDefinition*. Its values are defined in Table 34A.

Table 34A – StructureType Values

Name	Value	Description
Structure	0	A *Structure* without optional fields where none of the fields allow subtyping
StructureWithOptionalFields	1	A *Structure* with optional fields where none of the fields allow subtyping
Union	2	A *Union DataType* where none of the fields allow subtyping
StructureWithSubtypedValues	3	A Structure without optional fields where one or more of the fields allow subtyping
UnionWithSubtypedValues	4	A *Union DataType* where one or more of the fields allow subtyping

Replace Table 36 with the following:

Name	Type	Description
StructureField	Structure	
name	String	A name for the field that is unique within the *StructureDefinition*.
description	LocalizedText	A localized description of the field
dataType	NodeId	The *NodeId* of the *DataType* for the field. When used by a StructureDefinition with a structureType of *Structure,* StructureWithOptionalFields or Union then the datatype shall be a concrete DataType, *BaseDataType DataType* or *Structure DataType* *If the structureType is* StructureWithSubtypedValues, or UnionWithSubTypedValues and the isOptional field is true then the encoding of the field shall be able to transport any subtype of the DataType. Part 6 provides more details of specific encodings.
valueRank	Int32	The value rank for the field. It shall be Scalar (-1) or a fixed rank Array (>=1).
arrayDimensions	UInt32[]	This field specifies the maximum supported length of each dimension. If the maximum is unknown the value shall be 0. The number of elements shall be equal to the value of the v*alueRank field*. This field shall be null if v*alueRank* ≤ 0. The maximum number of elements of an array transferred on the wire is 2147483647 (max Int32).

	maxStringLength	UInt32	If the dataType field is a String or ByteString then this field specifies the maximum supported length. If the maximum is unknown the value shall be 0. If the dataType field is not a String or ByteString the value shall be 0. If the valueRank is greater than 0 this field applies to each element of the array.
	isOptional	Boolean	If the structureType is StructureWithOptionalFields this field indicates if a data type field in a *Structure* is optional. In this case a value of FALSE means the *StructureField* is always present in all occurances of the *Structure DataType and a value of* TRUE means the *StructureField* may be present in an occurance of the *Structure DataType*. If the *structureType* is *Structure* or *Union* this field shall be FALSE and shall be ignored. If the structureType is StructureWithSubtypedValues, or UnionWithSubTypedValues this field is used to indicate if the data type field allows subtyping. Subtyping is allowed when set to TRUE.

Topic	*Applicability of MaxStringLength Property*
Errata Version	1.04.10
Spec Reference	Part 3 5.6.2 Variable NodeClass, 5.6.4 DataVariable and 8.51 StructureField
Mantis Reference	0006479
Problem Statement	The MaxStringLength Property is defined for Strings but not for LocalizedText which it should be.
Solution	In Table 13 replace the description of MaxStringLength with: *Only used for DataVariables having a String or* a *LocalizedText DataType (the text* field*).* *This optional Property indicates the maximum number of bytes supported by the DataVariable.* In 5.6.4 replace the description of the MaxStringLength Property with: The *Property MaxStringLength* indicates the maximum number of bytes of a *String* or the *text* field of a *LocalizedText* value. If a *Server* does not impose a maximum number of bytes or is not able to determine the maximum number of bytes this *Property* shall not be provided. If this *Property* is provided, then the *MaxCharacters Property* shall not be provided. In 5.6.4 replace the description of the MaxCharacters Property with: The *Property MaxCharacters* indicates the maximum number of Unicode characters of a *String* or the *text* field of a *LocalizedText* value. If a *Server* does not impose a maximum number of Unicode characters or is not able to determine the maximum number of Unicode characters this *Property* shall not be provided. If this *Property* is provided then the *MaxStringLength Property* shall not be provided. In Table 36 replace the description of the maxStringLength with: If the dataType field is a String, LocalizedText (text field) or ByteString then this field specifies the maximum supported length in bytes. If the maximum is unknown the value shall be 0. If the dataType field is not a String or ByteString the value shall be 0. If the valueRank is greater than 0 this field applies to each element of the array.

Topic	*inverse references to built-in DataType*
Errata Version	1.04.10
Spec Reference	Part 3 5.8.3 DataType NodeClass
Mantis Reference	0005824
Problem Statement	No mandatory inverse HasSubtype and HasEncoding reference to built-in DataType.
Solution	In 5.8.3 replace the paragraph describing HasSubtype with: *HasSubtype References* may be used to expose a data type hierarchy in the *AddressSpace*. The semantic of subtyping is only defined to the point, that a Server may provide instances of the subtype instead of the DataType. *Clients* should not make any assumptions about any other semantic with that information. For example, it might not be possible to cast a value of one data type to its base data type. *Servers* need not provide *HasSubtype References*, even if their *DataTypes* span a type hierarchy, however it is required that the subtype provides the inverse *Reference* to its supertype. Some restrictions apply for subtyping enumeration DataTypes as defined in 8.14. In 5.8.3 replace the paragraph describing HasEncoding with: *HasEncoding References* point from the *DataType* to its *DataTypeEncodings*. Each concrete *Structured DataType* can point to many *DataTypeEncodings*, but each *DataTypeEncoding* shall belong to one *DataType*, that is, it is not permitted for two *DataType Nodes* to point to the same *DataTypeEncoding Object* using *HasEncoding References*. The *DataTypeEncoding Node* shall provide the inverse *HasEncoding Reference* to its *DataType*.

Topic	Remove the concept of NamingRule
Errata Version	1.04.11
Spec Reference	Part 3 5.5.1 Object NodeClass 6.4.4.2 Properties describing ModellingRules 6.4.4.3 Subtyping Rules for Properties of ModellingRules 6.4.4.5.1 Titles of Standard ModellingRules 6.4.4.5.2 Mandatory 6.4.4.5.3 Optional 8.29 NamingRuleType
Mantis Reference	0007907
Problem Statement	The NamingRule concept is flawed and should be removed.

Solution

In 5.5.1
Remove the row “NamingRule” from Table 11.
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 20 specifies the allowed changes on the Properties when overriding the ModellingRules in the subtype.

Table 20 - 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	1.04.1
Spec Reference	Part 4 7.36.2 Token Encryption and Proof of Possession
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 guess the server private key. The issue is described CVE-2018-7559.
Solution	Add the following statements: 7.36.2.1 Overview To prevent the leakage of information useful to attackers, *Servers* shall ensure the process of validating *UserIdentityTokens* completes in a fixed interval whether an error occurs or not. The process of validation includes decrypting, checking for padding and checking for a valid nonce. If any errors occur the return code is *Bad_IdentityTokenInvalid*. *Servers* shall log details of any failure to validate a *UserIdentityToken* and shall lock out *Client* applications with multiple failures (5 or so). 7.36.2.2 Legacy Encrypted Token Secret Format 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	1.04.2
Spec Reference	Part 4 7.36.2 Token Encryption and Proof of Possession
Mantis Reference	0004231
Problem Statement	User credentials can be compromised if a client does not validate a server certificate before sending an encrypted UserIdentityToken. The issue is described CVE-2018-12087.
Solution	Add the following statement: 7.36.2.1 Overview *Clients* shall validate the *Server Certificate* and ensure it is trusted before sending a *UserIdentityToken* encrypted with the *Certificate*.

Topic	BrowseDirection is missing the option INVALID
Errata Version	1.04.5
Spec Reference	Part 4 Table 34 – Browse Service Parameters
Mantis Reference	0005343
Problem Statement	The BrowseDirection has INVALID_3 in the Nodeset and code but not in the specification.
Solution	Table 34 – Browse Service Parameters Parameter browseDirection Description Add the following option: INVALID_3 no value specified

Topic	QueryFirst parameter includeSubTypes inconsistency with Nodeset
Errata Version	1.04.5
Spec Reference	Part 4 Table 47 – QueryFirst Request Parameters
Mantis Reference	0005344
Problem Statement	The casing of includeSubTypes is inconsistent between Nodeset / code and specification
Solution	Table 47 – QueryFirst Request Parameters Replace parameter name includeSubtypes with includeSubTypes

Topic	SimpleAttributeOperand structure field name typeId inconsistency with Nodeset
Errata Version	1.04.5
Spec Reference	Part 4 Table 1 – SimpleAttributeOperand
Mantis Reference	0005345
Problem Statement	The field name in the specification is typeId and the name in the Nodeset is typeDefinitionId
Solution	Table 2 – SimpleAttributeOperand Replace parameter name typeId with typeDefinitionId

Topic	AggregateFilterResult structure is missing revisedAggregateConfiguration
Errata Version	1.04.5
Spec Reference	Part 4 Table 146 – AggregateFilterResult structure
Mantis Reference	0005212
Problem Statement	AggregateFilterResult structure field revisedAggregateConfiguration is in the Nodeset / code but not in the specification
Solution	Table 146 – AggregateFilterResult structure Add field revisedAggregateConfiguration with type AggregateConfiguration and description The actual aggregateConfiguration that the Server shall use. The structure is defined in Table 145.

Topic	GenericAttributes structure is missing fields
Errata Version	1.04.5
Spec Reference	Part 4 Table 158 – GenericAttributes
Mantis Reference	0005346
Problem Statement	GenericAttributes structure fields are in the Nodeset / code but not in the specification
Solution	Table 158 – GenericAttributes Replace table with the following table

Name	Type	Description
GenericAttributes	structure	Defines a generic structure for passing in any number of *Attributes.*
specifiedAttributes	UInt32	A bit mask that indicates which fields contain valid values. A field shall be ignored if the corresponding bit is set to 0. The bit values are defined in Table 149.
displayName	LocalizedText	See Part 3 for the description of this *Attribute*.
description	LocalizedText	See Part 3 for the description of this *Attribute*.
writeMask	UInt32	See Part3 for the description of this *Attribute*.
userWriteMask	UInt32	See Part 3 for the description of this *Attribute*.
attributeValues	GenericAttributeValue []	Defines one *attributeId* and *value* combination.
attributeId	IntegerId	Id of the *Attribute* specified.
value	BaseDataType	Value of the *Attribute* specified.

Topic	TimestampsToReturn is missing the option INVALID
Errata Version	1.04.5
Spec Reference	Part 4 Table 179 – TimestampsToReturn values
Mantis Reference	0005347
Problem Statement	Description The TimestampsToReturn has INVALID_4 in the Nodeset and code but not in the specification.
Solution	Table 179 – TimestampsToReturn values Add the following enumeration option to the table Value: INVALID_4 Description: No value specified.

Topic	UserIdentityTokenType enumeration data type mismatch with Nodeset
Errata Version	1.04.5
Spec Reference	Part 4 Table 190 – UserTokenPolicy
Mantis Reference	0005348
Problem Statement	The enumeration type name in the Nodeset / code is UserTokenType and it is UserIdentityTokenType in the specification
Solution	Table 190 – UserTokenPolicy Change enumeration type name in the table from UserIdentityTokenType to UserTokenType.

Topic	Determining if a Certificate is Trusted
Errata Version	1.04.8
Spec Reference	Part 4 6.1.3 Determining if a Certificate is Trusted
Mantis Reference	0004666
Problem Statement	Determining if a Certificate is Trusted is not consistent with Part 12

Solution	6.1.3 Determining if a Certificate is Trusted Replace the following first three paragraphs *Applications* shall never communicate with another application that they do not trust. An *Application* decides if another application is trusted by checking whether the *Application Instance Certificate* for the other application is trusted. Applications shall rely on lists of *Certificates* provided by the *Administrator* to determine trust. There are two separate lists: a list of trusted *Applications* and a list of trusted *Certificate Authorities* (CAs). If an application is not directly trusted (i.e. its *Certificate* is not in the list of trusted applications) then the application shall build a chain of *Certificates* back to a trusted CA. When building a chain each *Certificate* in the chain shall be validated. If any validation error occurs then the trust check fails. Some validation errors are non-critical which means they can be suppressed by a user of an *Application* with the appropriate privileges. Suppressed validation errors are always reported via auditing (i.e. an appropriate Audit event is raised). Building a trust chain requires access to all *Certificates* in the chain. These *Certificates* may be stored locally or they may be provided with the application *Certificate*. Processing fails with Bad_SecurityChecksFailed if a CA *Certificate* cannot be found. With *Applications* shall never communicate with another application that they do not trust. An *Application* decides if another application is trusted by checking whether the *Application Instance Certificate* for the other application is trusted. A *Certificate* is only trusted if its chain can be validated. Applications shall rely on lists of *Certificates* provided by the *Administrator* to determine trust. There are two separate lists: a list of trusted *Certificates* and a list of issuer *Certificates* (i.e. CAs). The list of trusted *Certificates* may contain a *Certificate* issued to another *Application* or it may be a *Certificate* belonging to a CA. The list of issuer *Certificates* contains *CA Certificates* needed for chain validation that are not in the list of trusted *Certificates.* When building a chain each *Certificate* in the chain shall be validated back to a CA with a self-signed *Certificate* (a.k.a. a root CA). If any validation error occurs then the trust check fails. Some validation errors are non-critical which means they can be suppressed by a user of an *Application* with the appropriate privileges. Suppressed validation errors are always reported via auditing (i.e. an appropriate Audit event is raised). Determining trust requires access to all *Certificates* in the chain. These *Certificates* may be stored locally or they may be provided with the application *Certificate*. Processing fails with Bad_SecurityChecksFailed if an element in the chain cannot be found. A *Certificate* is trusted if the *Certificate* or at least one of the *Certificates* in the chain are in the list of trusted *Certificates* for the *Application* and the chain is valid.

Topic	Allow handling of partial IndexRange support
Errata Version	1.04.8
Spec Reference	Part 4 Table 61 – Write Operation Level Result Codes
Mantis Reference	0004154
Problem Statement	A server may be able to support a subset of the potential IndexRanges for Write
Solution	Table 61 – Write Operation Level Result Codes Bad_IndexRangeInvalid Add the following text to Description: It is also used if writing of *IndexRange* is supported in general for a *Node* but the passed *IndexRange* cannot be written by the Server.

Topic	Wrong keep alive counter handling
Errata Version	1.04.8
Spec Reference	Part 4 Table 85 – Subscription State Table
Mantis Reference	0005367
Problem Statement	The keep alive counter handling is wrong for case #9
Solution	Table 85 – Subscription State Table Add following Action to #9: KeepAliveCounter--

Topic	Order or fields in NodeAttributes Structures
Errata Version	1.04.8
Spec Reference	Part 4 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 Table 150 – ObjectAttributes Table 151 – VariableAttributes Table 152 – MethodAttributes Table 153 – ObjectTypeAttributes Table 153 – ObjectTypeAttributes Table 155 – ReferenceTypeAttributes Table 156 – DataTypeAttributes Table 157 – ViewAttributes Move structure fields 'writeMask' and 'userWriteMask' up in the list behind the structure field 'description'

Topic	Subscription retransmission queue
Errata Version	1.04.8
Spec Reference	Part 4 5.13 Subscription Service Set 5.13.1.1 Description 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 5.13.1.1 Description Replace i) with the following text: 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. 5.13.5 Publish Table 95 – Publish Service Parameters Replace availableSequenceNumbers description with the following text: 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). 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. Table 97 – Publish Operation Level Result Codes Add status code to table Good_RetransmissionQueueNotSupported The Server does not support retransmission queue and acknowledgement of sequence numbers is not available.

Topic	Clarified behaviour for ModifySubscription on Durable Subscriptions
Errata Version	1.04.10
Spec Reference	Part 4 6.8 Durable Subscriptions
Mantis Reference	0006310
Problem Statement	Behaviour of ModifySubscription for Durable Subscriptions is not defined
Solution	6.8 Durable Subscriptions Add following: The *revisedLifeTimeInHours* is used to set the *LifeTimeCount* of the *Subscription*. *ModifySubscription* can be used to change the parameters of the durable Subscription. If the *Client* would like to keep the previous life time setting, the *Client* needs to calculate the *LifeTimeCount* based on the *revisedLifeTimeInHours* and the *PublishingInterval*. *ModifySubscription* does not change the durable mode of the *Subscription*.

Topic	*AuditOpenSecureChannelEventType*
Errata Version	1.04.10
Spec Reference	Part 4 6.5.5 Auditing for SecureChannel Service Set
Mantis Reference	0006473
Problem Statement	Auditing of Certificate validation errors for the SecureChannel Service Set, can't use AuditEntryId to correlate the AuditCertificateEventType and AuditOpenSecureChannelEventType events. This is due to the AuditEntryId being in the encrypted body of the request and under error conditions can't be decrypted.
Solution	6.5.5 Auditing for SecureChannel Service Set Replace: In the case of Certificate validation errors the description should include the audit EventId of the specific AuditCertificateEventType that was generated to report the Certificate error. The AuditCertificateEventType shall also contain the detailed Certificate validation error. The additional parameters should include the details of the request. It is understood that these events may be generated by the underlying Communication Stacks in many cases, but they shall be made available to the Server and the Server shall report them. With: In the case of Certificate validation errors the CertificateErrorEventId of the AuditOpenSecureChannelEventType should include the audit EventId of the specific AuditCertificateEventType that was generated to report the Certificate error. The AuditCertificateEventType shall also contain the detailed Certificate validation error. The additional parameters should include the details of the request. It is understood that these events may be generated by the underlying Communication Stacks in many cases, but they shall be made available to the Server and the Server shall report them.

Topic	Clarified error handling for event monitored items
Errata Version	1.04.10
Spec Reference	Part 4 5.12.1.5 Queue parameters
Mantis Reference	0004159
Problem Statement	Part 3 defines SystemStatusChangeEventType for event monitored item errors but it is not referenced in Part 4
Solution	5.12.1.5 Queue parameters Replace last sentence It shall be handled like other connection error scenarios. With It shall be handled like other connection error scenarios using the SystemStatusChangeEventType with the ServerState COMMUNICATION_FAULT. For any fatal error during event processing like out of memory situations, the Server should queue an SystemStatusChangeEventType event with the ServerState COMMUNICATION_FAULT and the source set to the Server Object. If there are no resources available at the time the error happens, the Server should flag an error internally until there are resources to further process Events for the MonitoredItem.

Topic	Clarified StatusCode handling in Event fields
Errata Version	1.04.10
Spec Reference	Part 4 Table 119 – Basic FilterOperator definition Table 122 – Conversion rules
Mantis Reference	0004188
Problem Statement	StatusCodes indicating not existing values in Event fields need special handling
Solution	Table 119 – Basic FilterOperator definition Operator IsNull Add to Description: TRUE If the value in operand[0] is a StatusCode instead of the field DataType. Table 122 – Conversion rules Source Type StatusCode Change all implicit conversion to explicit conversions

Topic	Clarified TransferSubscription verification requirements
Errata Version	1.04.12
Spec Reference	Part 4 5.3.7 TransferSubscription
Mantis Reference	0007591
Problem Statement	Server is not able to verify the Client Profiles
Solution	5.3.7.1 Description Remove 'and that the potentially new Client supports the Profiles that are necessary for the Subscription'

Topic	IssuedIdentityToken expiration behaviour
Errata Version	1.04.12
Spec Reference	Part 4 7.36.6 IssuedIdentityToken
Mantis Reference	0007199
Problem Statement	IssuedIdentityToken expiration behaviour is not defined
Solution	7.36.6 IssuedIdentityToken Add the following paragraph: *IssuedIdentityTokens* have an expiration time, and a *Server* shall invalidate the credentials of the *Session* within a configurable time after the token expires. The *Session* shall stay valid with an *Anonymous* user token if the *Server* allows *Anonymous* users. *Clients* should renew the token with *ActivateSession* before the expiration time to avoid communication interruption.

Topic	UserTokenPolicy PolicyId null value
Errata Version	1.04.12
Spec Reference	Part 4 7.37 UserTokenPolicy
Mantis Reference	0008130
Problem Statement	UserTokenPolicy PolicyId null value behaviour not defined
Solution	7.37 UserTokenPolicy policyId Add the following clarification: The identifier may be null or empty. Null or empty are equal. The identifier shall be unique across the *UserTokenPolicies* assigned by the *Server*. 7.36.3 AnonymousIdentityToken 7.36.4 UserNameIdentityToken 7.36.5 X509IdentityToken 7.36.6 IssuedIdentityToken policyId Add the following clarification: Servers that provide a null or empty *PolicyId* shall accept null or empty and treat them as equal.

Topic	Certificate validation
Errata Version	1.04.12
Spec Reference	Part 4 Table 106 – Certificate validation steps
Mantis Reference	0004704
Problem Statement	No detailed error should be returned for Find Revocation List error
Solution	Table 106 – Certificate validation steps Added to description for Find Revocation List: Bad_SecurityChecksFailed should be reported back to the Client.

Topic	Requirements for Session-less Service invocation
Errata Version	1.04.12
Spec Reference	Part 4 6.3 Session-less Service invocation 6.3.1 Description
Mantis Reference	0008756
Problem Statement	The requirement in Session-less Service invocation implies that the feature must be implemented as soon as the listed services are implemented.
Solution	6.3 Session-less Service invocation 6.3.1 Description Replace All Services belonging to these Service Sets that are supported by a Server via a Session shall also be supported via the SessionlessInvoke Service. With If Session-less Service invocation is supported by a Server, all Services belonging to these Service Sets that are supported by a Server via a Session shall also be supported via the SessionlessInvoke Service.

Topic	Server certificates in GetEndpoints
Errata Version	1.04.12
Spec Reference	Part 4 5.4.4 GetEndpoints 5.4.4.1 Description
Mantis Reference	0008982
Problem Statement	It is only explicitly mentioned when the client should ignore the certificate in GetEndpoints response but not when the certificate is required.
Solution	5.4.4 GetEndpoints 5.4.4.1 Description Add If the securityPolicyUri is not None or one of the UserTokenPolicies requires encryption, the Server shall include the ApplicationInstanceCertificate in the EndpointDescription.

Topic	Check of domain name with ‘local’
Errata Version	1.04.12
Spec Reference	Part 4 5.5.2 OpenSecureChannel 5.5.2.1 Description
Mantis Reference	0006768
Problem Statement	Check of domain name with ‘local’ is not consistent with Part 12
Solution	5.5.2 OpenSecureChannel 5.5.2.1 Description Add Servers shall not append the 'local' top level domain to any domains declared in their Certificate; an unqualified domain name is used if a more appropriate qualifier does not exist. Clients using a URL returned from an LDS-ME shall remove the 'local' top level domain when checking the domain against the Server Certificate.

OPC UA Specification: Part 5 – Information Model

Topic	Correction to *AudioVariableType*
Errata Version	1.04.1
Spec Reference	Part 5 7.19 – *AudioVariableType*
Mantis Reference	0004181
Problem Statement	The *DataType* of the *AudioVariableType* is incorrectly specified.
Solution	The *DataType defined in Table 78 should be AudioDataType* and will be corrected in version 1.05.

Topic	Correction to DataTypeDescriptionType
Errata Version	1.04.2
Spec Reference	Part 5 D5.3 – DataTypeDescriptionType
Mantis Reference	0004316
Problem Statement	The *DataType* of the DataTypeDescriptionType is incorrectly specified.
Solution	The *DataType defined in Table D.3 should be String* and will be corrected in version 1.05.

Topic	Application authentication in RoleType::Identities and inconsistency in contained enumeration data type
Errata Version	1.04.6
Spec Reference	Part 5 Table F.3 – IdentityMappingRuleType
Mantis Reference	0005326, 0005554
Problem Statement	Application authentication only is not possible in RoleType::Identities. Inconsistency between enumeration data type defined for IdentityMappingRuleType and the one used in UANodeSet
Solution	Table F.3 – IdentityMappingRuleType Replace definition of criteriaType with the following definition: Type Enumeration IdentityCriteriaType Description: The type of criteria contained in the rule. Username_1 The rule specifies a UserName from a UserNameIdentityToken. Thumbprint_2 The rule specifies the Thumbprint of a User or CA Certificate. Role_3 The rule is a Role specified in an Access Token. GroupId_4 The rule is a user group specified in the Access Token. Anonymous_5 The rule specifies Anonymous UserIdentityToken. AuthenticatedUser_6 The rules specify any non-Anonymous UserIdentityToken. Application_7 The rule specifies the Application Instance Certificate of a Client. The criteria is the ApplicationUri from the Client Certificate which is trusted by the Server. This criteria type is used if a Role should be granted to a Session for Application Authentication with Anonymous user token. If instead a Role should be granted to a Session for Application Authentication combined with User Authentication, the Applications Property on the RoleType is used instead.

Topic	Correction to *ServerCapabilitiesType*
Errata Version	1.04.8
Spec Reference	Part 5 6.3.2 *ServerCapabiltiesType*
Mantis Reference	0004344
Problem Statement	*ServerProfileArray* content clarification.
Solution	Replaced description text of ServerProfileArray after Table 10 with the following: *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. Note: In Version 1.05 the String shall be the URI of the Profile.

Topic	ValueRank of SelectionListType
Errata Version	1.04.8
Spec Reference	Part 5 7.18 SelectionListType
Mantis Reference	0004264
Problem Statement	SelectionListType allowing its data type to be an array (ValueRank Any). However, the Property Selections is only a one-dimensional array.
Solution	In Table 77 replace the ValueRank Attribute Value -2 (-2 = Any) with -1 (-1 - Scalar).

Topic	NamespaceMetadataType DefaultAccessRestrictions Property
Errata Version	1.04.8
Spec Reference	Part 5 Table 21 NamespaceMetadataType Definition
Mantis Reference	0004319
Problem Statement	The Datatype of the DefaultAccessRestrictions Property is incorrect.
Solution	In Table 21 replace the DefaultAccessRestrictions Properties DataType of Uint16 with AccessRestrictionsType.

Topic	*FileTransferStateMachineType Reset Method*
Errata Version	1.04.10
Spec Reference	Part 5 Table C.17 FileTransferStateMachineType
Mantis Reference	0004070
Problem Statement	The Reset Method's Modelling Rule is missing.
Solution	In Table C.17 add the Modelling Rule Mandatory to the Reset Method.

Topic	*ServerType ServiceLevel Property*
Errata Version	1.04.10
Spec Reference	Part 5 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 9 with: *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	1.04.10
Spec Reference	Part 5 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 10 with: 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. Replace the definition of the *MaxStringLength Property* which follows after Table 10 with: The MaxStringLength Property indicates the maximum length of Strings supported by Variables, *Method Arguments* and *Event* fields of the Server. Servers may override this setting by adding the MaxStringLength Property defined in Part 3 to an individual DataVariable. If a Server does not impose a maximum number of bytes or is not able to determine the maximum number of bytes this Property shall not be provided.. Replace the definition of the *MaxByteStringLength Property* which follows after Table 10 with: The MaxByteStringLength Property indicates the maximum number of bytes in a ByteString supported by Variables, *Method Arguments* and *Event* fields of the Server. It also specifies the default maximum size of a FileType Object's read and write buffers. Servers may override this setting by adding the MaxByteStringLength Property defined in Part 3 to an individual DataVariable or FileType Object. If a Server does not impose a maximum number of bytes or is not able to determine the maximum number of bytes this Property shall not be provided.

Topic	*OptionSetType*
Errata Version	1.04.10
Spec Reference	Part 5 Table 76 OptionSetType Definition
Mantis Reference	0006320
Problem Statement	OptionSetType is defined with a ValueRank of Scalar and incorrectly includes the ArrayDimensions Attribute.
Solution	In Table 76 remove the ArrayDimensions Attribute.

Topic	*AuditEventType*
Errata Version	1.04.10
Spec Reference	Part 5 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: *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.

Topic	*AuditOpenSecureChannelEventType*
Errata Version	1.04.10 (Superseded by Errata 1.04.12 Mantis 8435)
Spec Reference	Part 5 6.4.6 AuditOpenSecureChannelEventType
Mantis Reference	0006470
Problem Statement	Auditing of Certificate validation errors for the SecureChannel Service Set, defined in Part 4, can't use AuditEntryId to correlate the AuditCertificateEventType and AuditOpenSecureChannelEventType events. This is due to the AuditEntryId being in the encrypted body of the request and under error conditions can't be decrypted.
Solution	Add the following row to the end of Table 27: Property | Variable | CertificateErrorEventId | String | PropertyType | Optional At the end of 6.4.3 insert: *CertificateErrorEventId* is the *EventId* of the corresponding *AuditCertificateEventType Event* used to report a certificate error.

Topic	*Applicability of MaxStringLength Property*
Errata Version	1.04.10
Spec Reference	Part 5 6.3.2 ServerCapabilityType
Mantis Reference	0006515
Problem Statement	The MaxStringLength Property is defined for Strings but not for LocalizedText which it should be.
Solution	In 6.3.2 replace the description of MaxStringLength with: *The MaxStringLength Property indicates the maximum number of bytes in Strings and LocalizedText supported by Variables, Method arguments and Event fields of the Server. Servers may override this setting by adding the MaxStringLength Property defined in OPC 10000-3 to an individual DataVariable. If a Server does not impose a maximum number of bytes or is not able to determine the maximum number of bytes this Property shall not be provided.*

Topic	*Creating an Instance*
Errata Version	1.04.10
Spec Reference	Part 5 6.4.2 Creating an Instance
Mantis Reference	0003979
Problem Statement	Optional Modeling Rule can cause broken behaviour of TranslateBrowsePathsToNodeId service when duplicated BrowseNames exists
Solution	In 6.4.2 replace the first 2 paragraphs with the follow 2 paragraphs: Instances inherit the initial values for the *Attributes* that they have in common with the *TypeDefinitionNode* from which they are instantiated, with the exceptions of the *NodeClass* and *NodeId*. When a *Server* creates an instance of a *TypeDefinitionNode* it shall create the same hierarchy of *Node*s beneath the new *Object* or *Variable* depending on the *ModellingRule* of each *InstanceDeclaration*. Standard *ModellingRules* are defined in 6.4.4.4. The *Node*s within the newly created hierarchy may be copies of the *InstanceDeclaration*s, the *InstanceDeclaration* itself or another *Node* in the *AddressSpace* that has the same *TypeDefinitionNode* and *BrowseName*. If new copies are created, then the *Attribute* values of the *InstanceDeclarations* are used as the initial values. An instance shall not be a *SourceNode* of a hierarchical *Reference* that has the same *BrowsePath* as an *InstanceDeclaration* of its *TypeDefinintion* which has an optional or mandatory *ModellingRule* except for the one based on the *InstanceDeclaration.*

Topic	WssIdentityToken should be IssuedIdentityToken
Errata Version	1.04.10
Spec Reference	Part 5 12.3 DataTypes defined in Part 5
Mantis Reference	0006740
Problem Statement	The name of IssuedIdentityToken was incorrectly stated as WssIdentityToken.
Solution	In 12.3 replace Table 130 with:

BrowseName
AnonymousIdentityToken
DataValue
DiagnosticInfo
ExpandedNodeId
SignedSoftwareCertificate
UserIdentityToken
UserNameIdentityToken
X509IdentityToken
IssuedIdentityToken
SecurityTokenRequestType
AddNodesItem
AddReferencesItem
DeleteNodesItem
DeleteReferencesItem
NumericRange
MessageSecurityMode
ApplicationDescription

In 12.3 replace Table 131 with:

Attributes	Value		
BrowseName	UserIdentityToken		
IsAbstract	TRUE		
References	**NodeClass**	**BrowseName**	**IsAbstract**
HasSubtype	DataType	UserNameIdentityToken	FALSE
HasSubtype	DataType	X509IdentityToken	FALSE
HasSubtype	DataType	IssuedIdentityToken	FALSE
HasSubtype	DataType	AnonymousIdentityToken	FALSE

Topic	*File Transfer*
Errata Version	1.04.10
Spec Reference	Part 5 4.2 FileType
Mantis Reference	0004648
Problem Statement	The OpenWithMasks method makes the file size specific to a particular file handle. The size property does not consider the file handle so it cannot provide an accurate value for file size
Solution	In 4.2.1 replace the definition of the *Size Property*, which is the first paragraph following Table 1, with the follow paragraph: *Size* defines the size of the file in Bytes. When a file is opened for write the size might not be accurate. If the *Server* can not accurately determine the size of the file, the Size *Property* shall be returned to a *Client* with a *StatusCode* of *Bad_NotSupported*.

Topic	*StateMachine HasEffect ReferenceType*
Errata Version	1.04.11
Spec Reference	Part 5 B4.14 HasEffect
Mantis Reference	0007254
Problem Statement	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.
Solution	Replace B4.14 with the following: The *HasEffect ReferenceType* is a concrete *ReferenceType* and can be used directly. It is a subtype of *NonHierarchicalReferences*. 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*. 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*. 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. The representation of the *HasEffect ReferenceType* in the *AddressSpace* is specified in B.13. **Table B.13 – HasEffect ReferenceType** **Attributes** / **Value** BrowseName: HasEffect InverseName: MayBeEffectedBy Symmetric: False IsAbstract: False **References** / **NodeClass** / **BrowseName** / **Comment**

Topic	*DataTypeDictionaryType*
Errata Version	1.04.11
Spec Reference	Part 5 D.5.2 DataTypeDictionaryType
Mantis Reference	0005788
Problem Statement	The Description of the optional property 'Deprecated' is not clear
Solution	In D.5.2 replace the definition of the *Deprecated Property* with: The *Deprecated Property* is used to indicate that all of the *DataType* definitions represented by the *DataTypeDictionaryType* are available through a *DataTypeDefinition Attribute*. Servers that provide *DataType* definitions as a *DataTypeDefinition Attribute* and through a *DataTypeDictionaryType* shall expose this *Property*. The value of this property should be set to True however it is only the presence of this optional *Property* which indicates the *Server* provides all *DataTypes* through *DataTypeDefinition Attributes.*

Topic	*Special Restrictions on subtyping StateMachines*
Errata Version	1.04.11
Spec Reference	Part 5 B.4.18 Special Restrictions on subtyping StateMachines
Mantis Reference	0005683, 0007774
Problem Statement	Clarification on Subtyping StateMachines needed
Solution	Replace B.4.18 with the following: In general, all rules on subtyping apply for *StateMachine* types as well. Some additional rules apply for *StateMachine* types. *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, HasGuard*) 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. 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	1.04.11
Spec Reference	Part 5 B.4.8 StateType 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: *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. In B.4.10 replace the first paragraph with the following: *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.

Topic	*AuditOpenSecureChannelEventType*
Errata Version	1.04.12
Spec Reference	Part 5 6.4.6 AuditOpenSecureChannelEventType
Mantis Reference	0008435
Problem Statement	The *CertificateErrorEventId Property* is incorrectly described as a *String DataType*.
Solution	Replace the added row (by Errata Version 1.04.10) at the end of Table 27:

Property	Variable	CertificateErrorEventId	ByteString	PropertyType	Optional

Keep the insertion made by Errata Version 1.04.10 at the end of 6.4.3: *CertificateErrorEventId* is the *EventId* of the corresponding *AuditCertificateEventType Event* used to report a certificate error.

OPC UA Specification: Part 6 – Mappings

Topic	Certificate authority key length verification rules.
Errata Version	1.04.2
Spec Reference	Part 6 6.1 Security handshake
Mantis Reference	0004223
Problem Statement	There are no constraints on the key length of Issuer Certificates which could lead to IOP issues.
Solution	Add to 6.1 after Table 35: The *MinAsymmetricKeyLength* and *MaxAsymmetricKeyLength* are constraints that apply to all Issuer *Certificates* in addition to the *Certificate*. In addition, the key length of issued *Certificates* shall be less than or equal to the key length of the issuer *Certificate*.

Topic	OpcUa_BadSequenceNumberInvalid status code is not defined.
Errata Version	1.04.2
Spec Reference	Part 6 6.7.6 Verifying Message Security and 7.1.5 Error handling
Mantis Reference	0004200
Problem Statement	OpcUa_BadSequenceNumberInvalid needs to be added.
Solution	Add to 6.7.6: If the *SequenceNumber* is not valid, the receiver shall log a *Bad_SequenceNumberInvalid* error. Add to 7.1.5 Table 55: Bad_SequenceNumberInvalid: The sequence number on the message was not valid.

Topic	Allowable content of DataTypeDefinition in Part 6 - Annex F is unclear.
Errata Version	1.04.2
Spec Reference	Part 6 Table F.1 – DataTypeDefinition
Mantis Reference	0004369
Problem Statement	It is unclear whether fields from the supertype should be included in the DataTypeDefinition.
Solution	Add to description of fields parameter in Table F.2: This list does not include fields inherited from a base data type.

Topic	Undefined status code Bad_CertificateUnknown.
Errata Version	1.04.2
Spec Reference	Part 6 6.7.6 Verifying Message Security
Mantis Reference	0004574
Problem Statement	The specification requires the use of a StatusCode that is not defined in the autogenerated code.
Solution	Remove the code from the specification and require that implementors use Bad_CertificateInvalid instead.

Topic	Generated NodeSets had wrong Namespace for DataTypeDefinition.Names.
Errata Version	1.04.2
Spec Reference	Part 6 F.12 DataTypeDefinition
Mantis Reference	0004712
Problem Statement	The code generators omitted the namespace index so many of the published NodeSets have errors.
Solution	Change UANodeSet.xsd to make the Name optional. Change the code generators to put the namespace index in if Name is different from BrowseName but omit otherwise.

Topic	Messages can be removed during a rollover.
Errata Version	1.04.3
Spec Reference	Part 6 6.7.2.4 SequenceNumber
Mantis Reference	0004545
Problem Statement	When using SignOnly an attacker with control over the network could remove Chunks during a rollover and it would not be detected.
Solution	Replace text after Table 46 in 6.7.2.4 with the following: A *SequenceNumber* may not be reused for any *TokenId*. The *SecurityToken* lifetime should be short enough to ensure that this never happens; however, if it does the receiver shall treat it as a transport error and force a reconnect. The *SequenceNumber* shall start at 1 023 and monotonically increase for all *Messages* and shall wrap around when it is equal to 4 294 966 271 (UInt32.MaxValue – 1 024). The first number after the wrap around shall be 1 023. Note that this requirement means that a *SequenceNumber* does not reset when a new *TokenId* is issued. The *SequenceNumber* shall be incremented by exactly one for each *MessageChunk* sent. For backward compatibility, receivers shall accept *SequenceNumbers* less than 1 023 and greater than 4 294 966 271 provided they are in sequence. Administrators shall be able to disable this backward compatibility. Receivers shall log a warning when a rollover does not conform to the current specification.

Topic	Rollover calculation not backward compatible.
Errata Version	1.04.4
Spec Reference	Part 6 6.7.2.4 SequenceNumber
Mantis Reference	0005265
Problem Statement	Rollover calculation not backward compatible. This is also a correction to an errata published in 1.04.3 which specified incorrect upper limit of 4 294 966 271 instead 4 294 966 272.
Solution	**The complete changed text with all errata applied:** Replace text after Table 46 in 6.7.2.4 with the following: A *SequenceNumber* may not be reused for any *TokenId*. The *SecurityToken* lifetime should be short enough to ensure that this never happens; however, if it does the receiver shall treat it as a transport error and force a reconnect. The *SequenceNumber* shall start at 1 023 and monotonically increase for all *Messages* and shall wrap around when it is equal to 4 294 966 272 (UInt32.MaxValue – 1 023). The first number after the wrap around shall be 1 023. Note that this requirement means that a *SequenceNumber* does not reset when a new *TokenId* is issued. The *SequenceNumber* shall be incremented by exactly one for each *MessageChunk* sent. For backward compatibility, receivers shall accept *SequenceNumbers* less than 1 023 and greater than 4 294 966 272 provided they are in sequence. Administrators shall be able to disable this backward compatibility. Receivers shall log a warning when a rollover does not conform to the current specification.

Topic	Need to add the term "DefaultValue" for all datatypes.
Errata Version	1.04.7
Spec Reference	Part 6 Table 1 Built-in Types.
Mantis Reference	0004930
Problem Statement	Need to add the term "DefaultValue" for all datatypes.
Solution	**Add a 'DefaultValue' column to Table 1:**

ID	Name	Default Value	Description
1	Boolean	false	A two-state logical value (true or false).
2	SByte	0	An integer value between −128 and 127 inclusive.
3	Byte	0	An integer value between 0 and 255 inclusive.
4	Int16	0	An integer value between −32 768 and 32 767 inclusive.
5	UInt16	0	An integer value between 0 and 65 535 inclusive.
6	Int32	0	An integer value between −2 147 483 648 and 2 147 483 647 inclusive.
7	UInt32	0	An integer value between 0 and 4 294 967 295 inclusive.
8	Int64	0	An integer value between −9 223 372 036 854 775 808 and 9 223 372 036 854 775 807 inclusive.
9	UInt64	0	An integer value between 0 and 18 446 744 073 709 551 615 inclusive.
10	Float	0	An IEEE single precision (32 bit) floating point value.
11	Double	0	An IEEE double precision (64 bit) floating point value.
12	String	null	A sequence of Unicode characters.
13	DateTime	DateTime.MinValue (see 5.1.4)	An instance in time.
14	Guid	All zeros	A 16-byte value that can be used as a globally unique identifier.
15	ByteString	null	A sequence of octets.
16	XmlElement	null	An XML element.
17	NodeId	null	An identifier for a node in the address space of an OPC UA *Server*.
18	ExpandedNodeId	null	A NodeId that allows the namespace URI to be specified instead of an index.
19	StatusCode	Good	A numeric identifier for an error or condition that is associated with a value or an operation.
20	QualifiedName	null	A name qualified by a namespace.
21	LocalizedText	null	Human readable text with an optional locale identifier.
22	ExtensionObject	null	A structure that contains an application specific data type that may not be recognized by the receiver.
23	DataValue	All fields set to default.	A data value with an associated status code and timestamps.
24	Variant	Empty	A union of all of the types specified above.
25	DiagnosticInfo	All fields set to default.	A structure that contains detailed error and diagnostic information associated with a StatusCode.

Topic	Protocol Version explanation contradicting.
Errata Version	1.04.7
Spec Reference	Part 6 6.7.4 OpenSecureChannel, Table 51 Hello Message and Table 52 Acknowledge Message.
Mantis Reference	0005267
Problem Statement	Protocol Version explanation contradicting.
Solution	**Replace ProtocolVersion description in Table 51 with:** The version of the UACP protocol requested by the Client. If Server does not support the requested version or any lower version it rejects the Client by returning Bad_ProtocolVersionUnsupported. If the Server supports the requested version or a lower version it shall return the version it will use in the Acknowledge Message. The ProtocolVersion for this version of the standard is 0. **Replace ProtocolVersion description in Table 52 with:** A protocol version supported by the Server that is less than or equal to the protocol version requested in the Hello Message. If the Client accepts the protocol version it shall ensure that it sends Messages that conform to this version. The ProtocolVersion for this version of the standard is 0.

Topic	Conflict between UA Spec and RFC 6960.
Errata Version	1.04.10
Spec Reference	Part 6 Table E.6 – CertificateValidationOptions
Mantis Reference	0006384
Problem Statement	Table E.6 says use crlDistributionPoint extension but RFC 6960 says use authorityInformationAccess extension.
Solution	**Replace CheckRevocationStatusOnline description in Table E.6 with:** Check the revocation status online. If set, the validator will look for the the authorityInformationAccess extension to find an OCSP (RFC 6960) endpoint which can be used to determine if the *Certificate* has been revoked. If the OCSP endpoint is not reachable then the validator will look for offline CRLs if the *CheckRevocationStatusOffine* bit is set. Otherwise, validation fails. This option is specified for *Issuer Certificates* and used when validating *Certificates* issued by that *Issuer*.

Topic	ECC key usage rules need to be specified.
Errata Version	1.04.10
Spec Reference	Part 6 6.2.2 Application Instance Certificate
Mantis Reference	0006474
Problem Statement	ECC key usage rules need to be specified.
Solution	**Replace keyUsage description in Table 38 with:** Specifies how the Certificate key may be used. For RSA keys, the keyUsage shall include digitalSignature, nonRepudiation, keyEncipherment and dataEncipherment. For ECC keys, the keyUsage shall include digitalSignature. Other keyUsage uses are allowed but not recommended. **Replace extendedKeyUsage description in Table 52 with:** Specifies additional limits on how the Certificate key may be used. For RSA keys, the extendedKeyUsage shall specify serverAuth and/or clientAuth. For ECC keys, the extendedKeyUsage may specify serverAuth and/or clientAuth. Other extendedKeyUsage bits are allowed.

Topic	Handling of StructureFields with abstract DataType.
Errata Version	1.04.10
Spec Reference	Part 6 F.13 DataTypeField
Mantis Reference	0006478
Problem Statement	Need a way to handle StructureFields with abstract DataType.
Solution	**Add AllowSubtypes to Table F.13:** AllowSubtypes | Boolean | This field is ignored if the *DataType* is *BaseDataType, Structure* or if the *DataType* has a fixed encoding such as a *String* or *Int32*. This field only applies to fields with a *DataType* which could have subtypes that have different encodings. This field shall be TRUE for all abstract *DataTypes* where the possible subtypes have different encodings. When TRUE, the field's value is allowed to contain subtypes of the *DataType*. All subtypes of *Structure* are encoded as an *ExtensionObject* (see 5.1.5). All *DataTypes* that are not subtypes of *Structure* are encoded as a *Variant* (see 5.1.6). The default value is false.

Topic	AllowSubtypes vs AllowSubTypes inconsistency
Errata Version	1.04.11
Spec Reference	Part 6 F.13 DataTypeField
Mantis Reference	0007381
Problem Statement	Casing specified in 1.04.10 errata was wrong.
Solution	**Modify row in Table F.13:** AllowSubTypes | Boolean | This field is ignored if the *DataType* is *BaseDataType, Structure* or if the *DataType* has a fixed encoding such as a *String* or *Int32*. This field only applies to fields with a *DataType* which could have subtypes that have different encodings. This field shall be TRUE for all abstract *DataTypes* where the possible subtypes have different encodings. When TRUE, the field's value is allowed to contain subtypes of the *DataType*. All subtypes of *Structure* are encoded as an *ExtensionObject* (see 5.1.5). All *DataTypes* that are not subtypes of *Structure* are encoded as a *Variant* (see 5.1.6). The default value is false.

Topic	No way to specify XmlSchemaUri in NodeSet.
Errata Version	1.04.11
Spec Reference	Part 6 F.2 UANodeSet
Mantis Reference	0007442
Problem Statement	UANodeSet needs to explain association with XSDs.
Solution	**Add row in Table F.1:** XmlSchemaUri | String | The URI for the XML schema namespace used to serialize values of the *DataTypes* defined by the Model. The field is required if *DataTypes* are defined in the *UANodeSet*. The UA XML Encoding (5.3) rules implicitly define the XML schema for any *DataType* described by a *DataTypeDefinition* (F.12).

Topic	Requirement on CA Flag for self-signed certificates is a potential security risk.
Errata Version	1.04.12

Spec Reference	Part 6 6.2.2 Application Instance Certificate
Mantis Reference	0008370, 0008670
Problem Statement	The basicContraints field in X.509 v3 Certificates needs to described.
Solution	**Add row in Table 36:** basicConstraints | (No mapping) | The *basicConstraints* field is completely described in RFC 5280. The *cA* flag Identifies whether the subject of the *Certificate* is a CA The *pathLength* specifies the maximum number of intermediate CAs in valid chains that follow this *Certificate*. The basicConstraints extension shall be present and shall not be ignored. The extension shall be validated and marking the extension as critical has no effect. For backward interoperability, any error related to the critical mark produced by software libraries shall be suppressed and logged as a warning. The *cA* flag shall be FALSE for any *ApplicationInstance Certificate*, however, TRUE shall be accepted to ensure backward interoperability when validating *ApplicationInstance Certificates* if revocation checks are enabled. If revocation checks are disabled then a *Certificate* with the cA flag set to TRUE should not be accepted. It should be possible to disable backward interoperability in configuration. If the cA flag is TRUE for a self-signed *ApplicationInstance Certificate,* then the pathLength should be 0. If an application accepts an *ApplicationInstance Certificate with cA* flag set to TRUE, it should write a warning to the log.

Topic	OpenSecureChannel Renew Issue with slow embedded devices.
Errata Version	1.04.12
Spec Reference	Part 6 6.7.4 – Establishing a SecureChannel
Mantis Reference	0007448
Problem Statement	6.7.4 forces embedded devices to block while processing asymmetric cryptographic operations.
Solution	**Replace Paragraph:** The *RevisedLifetime* tells the *Client* when it shall renew the *SecurityToken* by sending another *OpenSecureChannel* request. The *Client* shall continue to accept the old *SecurityToken* until it receives the *OpenSecureChannel* response. The *Server* has to accept requests secured with the old *SecurityToken* until that *SecurityToken* expires or until it receives a *Message* from the *Client* secured with the new *SecurityToken*. The *Server* shall reject renew requests if the *SenderCertificate* is not the same as the one used to create the *SecureChannel* or if there is a problem decrypting or verifying the signature. The *Client* shall abandon the *SecureChannel* if the *Certificate* used to sign the response is not the same as the *Certificate* used to encrypt the request. Note that datatype is a *UInt32* value representing the number of milliseconds instead of the *Double* (*Duration*) defined in Part 4. This optimization is possible because sub-millisecond timeouts are not supported. **With:** The *RevisedLifetime* tells the *Client* when it shall renew the *SecurityToken* by sending another *OpenSecureChannel* request. The *Client* shall continue to accept the old *SecurityToken* until it expires or it receives a *Message* from the *Server* secured with the new *SecurityToken*. The *Client* shall send a *Messages* secured with new *SecurityToken* as soon as it finishes processing the *OpenSecureChannel* response. The *Client* shall close the *SecureChannel* if the *Certificate* used to sign the response is not the same as the *Certificate* used to encrypt the request. The *Server* shall accept requests secured with the old *SecurityToken* until that *SecurityToken* expires or until it receives a *Message* from the *Client* secured with the new *SecurityToken*. The *Server* shall reject renew requests if the *SenderCertificate* is not the same as the one used to create the *SecureChannel* or if there is a problem decrypting or verifying the signature.

Topic	Decimal schema in Part 6 is quite different to the XSD.
Errata Version	1.04.12
Spec Reference	Part 6 5.3.2 Decimal
Mantis Reference	0007758
Problem Statement	The XML schema is inconsistent with definition of Decimal in 5.1.7.

Solution	**Replace XML example:**
	`<xs:complexType name="Decimal">`
	`  <xs:sequence>`
	`  <xs:element name="TypeId" type="tns:NodeId" minOccurs="0" />`
	`  <xs:element name="Body" minOccurs="0">`
	`  <xs:complexType>`
	`  <xs:sequence>`
	`  <xs:element name="Scale" type="xs:unsignedShort" />`
	`  <xs:element name="Value" type="xs:string" />`
	`  </xs:sequence>`
	`  </xs:complexType>`
	`  </xs:element>`
	`  </xs:sequence>`
	`</xs:complexType>`
	
	With:
	`<xs:complexType name="Decimal">`
	`  <xs:sequence>`
	`  <xs:element name="TypeId" type="tns:NodeId" minOccurs="0" />`
	`  <xs:element name="Body" minOccurs="0">`
	`  <xs:complexType>`
	`  <xs:sequence>`
	`  <xs:element name="Scale" type="xs:short" />`
	`  <xs:element name="Value" type="xs:string" />`
	`  </xs:sequence>`
	`  </xs:complexType>`
	`  </xs:element>`
	`  </xs:sequence>`
	`</xs:complexType>`

<table>
<tr><td>Topic</td><td>Clarification for UAInstance::ParentNodeId and validation of models.</td></tr>
<tr><td>Errata Version</td><td>1.04.12</td></tr>
<tr><td>Spec Reference</td><td>Part 6
Table F.7 – UAInstance</td></tr>
<tr><td>Mantis Reference</td><td>0007766</td></tr>
<tr><td>Problem Statement</td><td>ParentNodeId usage not defined clearly.</td></tr>
<tr><td>Solution</td><td>
<u>Replace Row in Table:</u>
<table>
<tr><td>ParentNodeId</td><td>NodeId</td><td>The NodeId of the Node that is the parent of the Node within the information model. This field is used to indicate that a tight coupling exists between the Node and its parent (e.g., when the parent is deleted, the child is deleted as well). This information does not appear in the AddressSpace and is intended for use by design tools.</td></tr>
</table>
<u>With:</u>
<table>
<tr><td>ParentNodeId</td><td>NodeId</td><td>The NodeId of the Node that is the parent of the Node within the information model. This field is used to indicate that a tight coupling exists between the Node and its parent (e.g., when the parent is deleted, the child is deleted as well). This information does not appear in the AddressSpace and is intended for use by design tools.
This field should be specified for all InstanceDeclarations and should reference a UANode which is the source of a HierarchicalReference to the Node.</td></tr>
</table>
</td></tr>
</table>

Topic	How can a tool find out which Namespace 0 NodeSet is newer / better?
Errata Version	1.04.12
Spec Reference	Part 6 Table F.1 – UANodeSet
Mantis Reference	0007979
Problem Statement	The PublicationDate is insufficient when comparing a 1.04 NodeSet to a 1.05 NodeSet.
Solution	**Add Row to Table:** ModelVersion | VersionString | The version of the *UANodeSet* expressed as a string which conforms to the syntax defined in the SemVer specification. The comparison rules from the specification are used to determine which *UANodeSet* precedes another. When the *ModelVersion* is specified in two *UANodeSets,* the *PublicationDate* is not used when comparing the *UANodeSets*. When the *ModelVersion* is specified in only one *UANodeSet,* the *PublicationDate* is used when comparing the *UANodeSets* and the *ModelVersion* is ignored. The *ModelVersion* is optional in previously released *UANodeSets* and it is required in all future *UANodeSets*.

Topic	Clarification on BaseDataType and Structure DataType in Structures.
Errata Version	1.04.12
Spec Reference	Part 6 Section 5.1 – DataEncoding General
Mantis Reference	0008442
Problem Statement	No description on what to do with Number, Integer and UInteger fields.
Solution	**Add Section 5.1.5a Number, Integer and UInteger:** *Number*, *Integer* and *UInteger* are abstract simple types defined in OPC 10000-3. When these types are used in *Structure* fields. the field value is encoded as a *Variant*.

OPC UA Specification: Part 7 – Profiles

The normative reference for Profiles, Facets and Conformance Units is now the online Profile Database (https://profiles.opcfoundation.org/v104/Reporting/), not Part 7. Henceforth Errata will not be published for Part 7 and any previously published Errata entries for Part 7 have been removed from this version.

OPC UA Specification: Part 8 – DataAccess

Topic	Inconsistent *DataTypes* in *EnumValues* and *MultiStateValueDiscreteType*
Errata Version	1.04.2
Spec Reference	Part 8 5.3.3.4 – MultiStateValueDiscreteType
Mantis Reference	0004195
Problem Statement	The *DataType* of *EnumValues* is *Int64*. The *MultiStateValueDiscreteType* however currently allows any subtype of *Number*. This could include *Float*, *Double*, and *Decimal*.
Solution	Only *DataTypes* that can be represented with *EnumValues* are allowed for *Variables* of *MultiStateValueDiscreteType*. These are Integers up to 64 Bits (signed and unsigned).

Topic	Parent Type for AnalogItemType
Errata Version	1.04.3
Spec Reference	Table 2 “AnalogItemType definition”
Mantis Reference	0004230
Problem Statement	The definition of *AnalogItemType* requires the *EURange Property*. This is quite inconvenient for certain applications or machine types.
Solution	Introduce an *AnalogBaseType* where all *Properties* are optional. Change the parent type for *AnalogItemType* from *DataItemType* to *AnalogBaseType*. *AnalogBaseType* and additional sub-types are defined in Amendment 1. Existing *Clients* will remain conformant to 1.04 even if they misinterpret the parent type of the *AnalogItemType*. Existing *Servers* will remain conformant to 1.04 even if they do not re-parent.

Topic	MultiStateValueDiscreteType is inconsistent with NodeSet
Errata Version	1.04.8
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". Add the following sentence to the description of properties below the table: “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	1.04.11
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: *ValuePrecision* specifies the maximum precision that the *Server* can maintain for the item based on restrictions in the target environment. *ValuePrecision* can be used for the following *DataTypes*: • 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. 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. • 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. • *ValuePrecision* can also be used for other subtypes of Double (like Duration) and other Number subtypes that can be represented by a Double. 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*. 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. 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	TestingConditionClassType vs. TestingConditionSubClassType
Errata Version	1.04.8
Spec Reference	Part 9 5.9.10 TestingConditionSubClassType
Mantis Reference	0004184
Problem Statement	There is an inconsistency between OPC UA Part 9 and UANodeSet In Part 9 the term is “TestingConditionSubClassType” In UANodeSet the term is “TestingConditionClassType”
Solution	Replace: TestingConditionSubClassType With: TestingConditionClassType In all occurances in Part 9.

Topic	Error code for Confirm method is incorrect
Errata Version	1.04.8
Spec Reference	Part 9 5.7.4 Confirm Method Table 33 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	Recommended localized names have various errors
Errata Version	1.04.8
Spec Reference	Part 9 Annex A – Multiple tables
Mantis Reference	0005585
Problem Statement	Table A.1 Latched state False State name is Latched, True State Name is Unlatched. These appear to be reversed as to what they should be. Table A.3 Latched state False State name is Verriegelt, True State Name is Entriegelt. These appear to be reversed as to what they should be. Table A.5 Latched state names are missing. Latched = Verrouillé Unlatched = Déverrouillé
Solution	Replace these rows in Table A.1

SilenceState	Silenced	Not Silenced
LatchedState	Latched	Unlatched
With		
SilenceState	Not Silenced	Silenced
LatchedState	Unlatched	Latched
In table A3 replace		
SilenceState	Stumm	Nicht Stumm
LatchedState	Verriegelt	Entriegelt
With		
SilenceState	Nicht Stumm	Stumm
LatchedState	Entriegelt	Verriegelt
In table A.5 replace		
SilenceState	Muette	Non-Muette
LatchedState		
With		
SilenceState	Non-Muette	Muette
LatchedState	Déverrouillé	Verrouillé

Topic	Clarification on abstractness of LimitAlarmType
Errata Version	1.04.8
Spec Reference	Part 9 5.8.11 LimitAlarmType
Mantis Reference	0004273
Problem Statement	The text describes the alarm type as Abstract, but the Table definition list the alarm type as not abstract.
Solution	Replace “The *LimitAlarmType* is an abstract type used to provide a base *Type* for *AlarmConditions* with multiple limits.” With “The *LimitAlarmType* is used to provide a base *Type* for *AlarmConditionTypes* with multiple limits”

Topic	AlarmConditionType Property AudibleSound (of AudioVariableType)
Errata Version	1.04.8
Spec Reference	Part 9 Table 35 – AlarmConditionType Definition.
Mantis Reference	0005213
Problem Statement	The AudibleSound variable is not a property and should be referenced with a HasComponent reference instead of a HasProperty ReferenceType
Solution	Replace the HasProperty reference for AudibleSound with a HasComponent reference.

Topic	SystemOffNormalAlarmType was intended for direct use but IsAbstract is True
Errata Version	1.04.10
Spec Reference	Part 9 Table 73 – 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 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>1.04.10</td></tr>
<tr><td>Spec Reference</td><td>Part 9
Table 19 – 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
<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	1.04.10
Spec Reference	Part 9 5.8.10 ShelvedStateMachineType 5.8.10.1 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 : "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. 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." With the following: "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. 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 ObjectId and MethodId for AlarmConditionType methods
Errata Version	1.04.11
Spec Reference	Part 9 Sections: 5.8.10.2 Unshelve Method 5.8.10.3 TimedShelve Method 5.8.10.4 OneShotShelve Method
Mantis Reference	0006486
Problem Statement	For *Servers* that do not expose *Alarm* instances, the use of *ObjectId* for *Method* calls is not clear.
Solution	Replace the text in the opening paragraph in each of the sections as indicated. Unshelve: "However, some *Servers* do not expose *Condition* instances in the *AddressSpace.* Therefore, all *Servers* shall also allow *Clients* to call the *Unshelve Method* by specifying *ConditionId* as the *ObjectId."* *with* "However, some *Servers* do not expose *Condition* instances in the *AddressSpace.* Therefore, all *Servers* shall also allow *Clients* to call the *Unshelve Method* by specifying *ConditionId* as the *ObjectId* where the *ConditionId* is the *Condition* that has *Shelving* child" *TimedShelve:* "However, some *Servers* do not expose *Condition* instances in the *AddressSpace.* Therefore, all *Servers* shall also allow *Clients* to call the *TimedShelve Method* by specifying *ConditionId* as the *ObjectId."* with "However, some *Servers* do not expose *Condition* instances in the *AddressSpace.* Therefore, all *Servers* shall also allow *Clients* to call the *TimedShelve Method* by specifying *ConditionId* as the *ObjectId* where the *ConditionId* is the *Condition* that has the *Shelving* child*."* OneShotShelve: "However, some *Servers* do not expose *Condition* instances in the *AddressSpace.* Therefore, all *Servers* shall also allow *Clients* to call the *OneShotShelve Method* by specifying *ConditionId* as the *ObjectId."* with "However, some *Servers* do not expose *Condition* instances in the *AddressSpace.* Therefore, all *Servers* shall also allow *Clients* to call the *OneShotShelve Method* by specifying *ConditionId* as the *ObjectId* where the *ConditionId* is the *Condition* that has the *Shelving* child*."*

Topic	Clarification needed for TwoStateVariableType TrueState and FalseState
Errata Version	1.04.11
Spec Reference	Part 9 Section 5.2 Two-state state machines Also sections: 5.5.2, 5.6.2, 5.7.2, 5.8.2, 5.8.13 (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 9 - ConditionType definition

Table 9.a – ConditionType Additional Subcomponents

rowsePath	References	NodeClass	BrowseName	DataType	TypeDefinition	Others
nabledState	HasProperty	Variable	TrueState	LocalizedText	PropertyType	X
nabledState	HasProperty	Variable	FalseState	LocalizedText	PropertyType	X

After Table 24 - DialogConditionType definition

Table 24.a– DialogConditionType Additional Subcomponents

rowsePath	References	NodeClass	BrowseName	DataType	TypeDefinition	Others
ialogState	HasProperty	Variable	TrueState	LocalizedText	PropertyType	X
ialogState	HasProperty	Variable	FalseState	LocalizedText	PropertyType	X

After Table 28 - AcknowledgeableConditionType definition

Table 28.a – AcknowledgeableConditionType Additional Subcomponents

rowsePath	References	NodeClass	BrowseName	DataType	TypeDefinition	Others
ckedState	HasProperty	Variable	TrueState	LocalizedText	PropertyType	X
ckedState	HasProperty	Variable	FalseState	LocalizedText	PropertyType	X
onfirmedState	HasProperty	Variable	TrueState	LocalizedText	PropertyType	X
onfirmedState	HasProperty	Variable	FalseState	LocalizedText	PropertyType	X

After Table 35 - AlarmConditionType table

Table 35.a – AlarmConditionType Additional Subcomponents

owsePath	References	NodeClass	BrowseName	DataType	TypeDefinition	Others
tiveState	HasProperty	Variable	TrueState	LocalizedText	PropertyType	X
tiveState	HasProperty	Variable	FalseState	LocalizedText	PropertyType	X
ppressedState	HasProperty	Variable	TrueState	LocalizedText	PropertyType	X
ppressedState	HasProperty	Variable	FalseState	LocalizedText	PropertyType	X
tOfServiceState	HasProperty	Variable	TrueState	LocalizedText	PropertyType	X
tOfServiceState	HasProperty	Variable	FalseState	LocalizedText	PropertyType	X
enceState	HasProperty	Variable	TrueState	LocalizedText	PropertyType	X
enceState	HasProperty	Variable	FalseState	LocalizedText	PropertyType	X
tchedState	HasProperty	Variable	TrueState	LocalizedText	PropertyType	X
tchedState	HasProperty	Variable	FalseState	LocalizedText	PropertyType	X

After Table 64 - NonExclusiveLimitAlarmType definition table

Table 64.a – NonExclusiveLimitAlarmType Additional Subcomponents

rowsePath	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	

Topic	AcknowledgeableConditionType the text in an error description references ConditionType instead of AcknowledgeableConditiontype
Errata Version	1.04.12
Spec Reference	Part 9 Table 30
Mantis Reference	0008421
Problem Statement	The acknowledgeableConditionType, acknowledged method and Confirm method error description table reference ConditionType instead of AcknowledgeableConditionType
Solution	Replace in both Table 30- Acknowledge result codes and in Table 33 - Confirm result codes the provided row Bad_NodeIdInvalid — Used to indicate that the specified *ObjectId* is not valid or that the *Method* was called on the ConditionType *Node*. See OPC 10000-4 for the general description of this result code. With Bad_NodeIdInvalid — Used to indicate that the specified *ObjectId* is not valid or that the *Method* was called on the AcknowledgeableConditionType *Node*. See OPC 10000-4 for the general description of this result code.

Topic	Definition of ConditionId need further clarification
Errata Version	1.04.12
Spec Reference	Part 9 5.5.3 Condition and branch instances
Mantis Reference	0009022
Problem Statement	The ConditionId is equivalent to the NodeId of an instance of an Alarm, but it is not clear that it needs to remain constant when the instance is not exposed in the AddressSpace
Solution	Update the fist paragraph in section 5.5.3 with the following text added to the end. “This *Condition* (and its representative *ConditionId)* follows the same rules associated with *Nodes* (and their representative *NodeIds*) in the *AddressSpace* (see OPC 10000-3). Therefore, once a *Condition* is created in a system that does not expose instances, any time that *Condition* is active it shall always have the same *ConditionId*. This allows higher level alarm management systems to operate on *Conditions* and to perform analysis of them.”

OPC UA Specification: Part 11 – Historical Access

Topic	Clarification of ReadDetails handling with ContinuationPoint.
Errata Version	1.04.12
Spec Reference	Part 11 section 6.3, 6.4.3.2, 6.4.3.3, 6.4.4.2, 6.4.5.2, 6.4.6.2
Mantis Reference	0007906
Problem Statement	The description of how to handle continuation point is inconsistent

Solution	In section 6.3 Replace paragraph: If the *Client* specifies a *ContinuationPoint*, then the *HistoryReadDetails* parameter and the TimestampsToReturn parameter are ignored, because it does not make sense to request different parameters when continuing from a previous call. It is permissible to change the *dataEncoding* parameter with each request. With: If the *Client* specifies a *ContinuationPoint*, then the *HistoryReadDetails* parameter and the *TimestampsToReturn* parameter shall be the same as in the original call. It is permissible to change the *dataEncoding* parameter with each request. If the *HistoryReadDetails* parameter or *TimestampsToReturn* parameter change the *Server* may ignore the updated parameters, it may detect any changes to the parameters and report an error or it may process the request as if it was a new request. The exact behaviour is *Server* specific. In section 6.4.3.2 Replace paragraph: If the request takes a long time to process then the *Server* can return partial results with a *ContinuationPoint*. This might be done if the request is going to take more time than the *Client* timeout hint. It may take longer than the *Client* timeout hint to retrieve any results. In this case the *Server* may return zero results with a *ContinuationPoint* that allows the *Server* to resume the calculation on the next *Client HistoryRead* call. With: The standard *ContinuationPoint* rules (see OPC 10000-4) apply. In addition, the following *ContinuationPoint* rule applies to *ReadRawModifiedDetails* If a *startTime*, *endTime* and *numValuesPerNode* are all provided and if more than *numValuesPerNode* values exist within that time range for a given *Node* then only *numValuesPerNode* values per *Node* shall be returned along with a *ContinuationPoint*. In section 6.4.3.3 Replace paragraph: If the request takes a long time to process then the *Server* can return partial results with a *ContinuationPoint*. This might be done if the request is going to take more time than the *Client* timeout hint. It may take longer than the *Client* timeout hint to retrieve any results. In this case the *Server* may return zero results with a *ContinuationPoint* that allows the *Server* to resume the calculation on the next *Client HistoryRead* call. With: The standard *ContinuationPoint* rules (see OPC 10000-4) apply. In addition, the following *ContinuationPoint* rule applies to *ReadRawModifiedDetails*. If more than *numValuesPerNode* values exist within that time range for a given *Node* then only *numValuesPerNode* values per *Node* are returned along with a *ContinuationPoint*. In section 6.4.4.2 Add the following to the end of the section:

	The standard *ContinuationPoint* rules (see OPC 10000-4) apply In section 6.4.5.2 Add the following to the end of the section: The standard *ContinuationPoint* rules (see OPC 10000-4) apply In section 6.4.6.2 Add the following to the end of the section: The standard *ContinuationPoint* rules (see OPC 10000-4) apply

Topic	Clarification of ReadDetails handling with ContinuationPoint.
Errata Version	1.04.12
Spec Reference	Part 11 section 4.3 Timestamps
Mantis Reference	0008534
Problem Statement	Inconsistent use of Bad_TimestampsToReturnInvalid versus Bad_TimestampNotSupported
Solution	In section 4.3 Replace paragraph: If a request is made requesting both *ServerTimestamp* and *SourceTimestamp* and the *Server* is only collecting the *SourceTimestamp* the *Server* shall return Bad_TimestampsToReturnInvalid. With: If a request is made requesting both *ServerTimestamp* and *SourceTimestamp* and the *Server* is only collecting the *SourceTimestamp,* the *Server* shall return Bad_TimestampNotSupported. Some *Historians* may aggregate data from underlying *Historians.* These *Servers* might have a mix of data, some with *ServerTimestamp* some without. The *Historian* may not know if a *HistoricalDataNode* supports *ServerTimestamps*. As a result of this uncertainty, the Bad_TimestampNotSupported may be returned on the service level or at the operation level.

Topic	Use of HistoricalDataConfigurationType
Errata Version	1.04.12
Spec Reference	Part 11 section 5.4 - Exposing supported functions and capabilities
Mantis Reference	0005574
Problem Statement	Historian configuration is defined but not mandatory, but for certain types of historical data retrievial, the client needs to understand how the data was collected. By providing a default configuration if no node specific configuration is available, a client can retrieve data.
Solution	Add the following section as section 5.4.3: **5.4.3 Default Configuration** This specification defines a manner of exposing the configuration associated with a *HistoricalNode*, but this configuration is not required for all nodes. If the configuration of individual nodes is not provided for all nodes, the *Historian* shall expose a default instance under the *Server Object.* If *HistoricalDataNodes* exist without a referenced configuration, an *Object* with a *BrowseName* of "*DefaultHAConfiguration*" shall be exposed. For any *Node* that does not have a referenced historical configuration but does support history, the default appropriate configuration shall be applied (*DefaultHAConfiguration* for *Variables)*. A *Client* shall assume the default configuration for any *HistoricalNode* that does not reference a historical configuration. The *DefaultHAConfiguration* object is defined in Table 8A. **Table 8A - DefaultHAConfiguration definition**

Attribute	**Value**			
BrowseName	DefaultHAConfiguration			
References	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**
OrganizedBy by the *Server* Object defined in 10000-5				
HasTypeDefinition	ObjectType	HistoricalDataConfigurationType	Defined in 5.2.2	

OPC UA Specification: Part 12 – Discovery and Global Services

Topic	AddCertificate handling of CA Certificates is problematic.
Errata Version	1.04.1
Spec Reference	Part 12 7.5.5 AddCertificate
Mantis Reference	0004172
Problem Statement	AddCertificate needs to clearly explain how CA certificates are added and how CRLs should be handled.
Solution	Replace: The *Client* shall provide the entire chain in the *certificate* argument (see Part 6) After validating the *Certificate*, the *Server* shall add the CA *Certificates* to the *Issuers* list in the *Trust List*. The leaf *Certificate* is added to the list specified by the *isTrustedCertificate* argument. With: This *Method* will return a validation error if the *Certificate* is issued by a CA and the *Certificate* for the issuer is not in the *Trust List*. This *Method* cannot provide CRLs so issuer Certificates cannot be added with this *Method*. Instead, CA *Certificates* and their CRLs shall be managed with the *Write Method* on the containing *TrustList Object.*

Topic	StartSigningRequest and CreateSigningRequest inconsistent.
Errata Version	1.04.1
Spec Reference	Part 12 7.6.3 StartSigningRequest and 7.7.6 CreateSigningRequest.
Mantis Reference	0004173
Problem Statement	StartSigningRequest and CreateSigningRequest have inconsistent requirements.
Solution	Replace: (certificateRequest parameter in 7.6.3): This blob shall include the *subjectAltName* extension that is in the *Certificate*. With: If the *CertificateRequest* is for an *ApplicationInstance Certificate* then it shall include all fields required by Part 6 such as the *subjectAltName*. Add: (certificateRequest parameter in 7.7.6): If the *CertificateRequest* is for an *ApplicationInstance Certificate* then it shall include all fields required by Part 6 such as the *subjectAltName*.

Topic	Subscribers should not register. Remove SUB server capability.
Errata Version	1.04.1
Spec Reference	Part 12 Annex D.
Mantis Reference	0004179
Problem Statement	There is no need for an application to discover a subscriber.
Solution	Remove: “SUB” capability from Table 71. Replace: Client applications that support the PUB or SUB capability can send or receive PubSub Messages but do not support the PubSub information model. With: Applications that support the PUB capability can send PubSub Messages but may not support the PubSub information model.

Topic	Inconsistency of null handling for certificateGroupId parameter in CertificateDirectoryType.
Errata Version	1.04.2
Spec Reference	Part 12 7.6.7 GetTrustList
Mantis Reference	0004171
Problem Statement	A default value needs to be specified.
Solution	Add to certificateGroupId argument description: If null, the CertificateManager shall return the trustListId for a suitable default group for the Application.

Topic	Requirements for CertificateManagers must be clarified.
Errata Version	1.04.2
Spec Reference	Part 12 7.1 CertificateManager Overview
Mantis Reference	0004198
Problem Statement	The requirements are not specific enough to be testable.
Solution	Replace: The CertificateManager shall support the following use cases: With: The CertificateManager should support the following features:

Topic	KeyCredentialManagement needs to support ECC.
Errata Version	1.04.2
Spec Reference	Part 12 8.4.4 StartRequest, 8.4.5 FinishRequest and 8.5.3 UpdateCredential.
Mantis Reference	0004299
Problem Statement	The mechanisms describing credential encryption are specific to RSA. They need to be made more general.
Solution	In 8.4.4 Replace: certificate: The *Certificate* containing the key used to encrypt the returned *KeyCredential* secret. This is the DER encoded form of an X.509 v3 *Certificate* as described in Part 6. Not specified if no encryption is required. If the *securityPolicyUri* is provided this field shall be provided. With: publicKey: A *Public Key* used to encrypt the returned *KeyCredential* secret. For RSA *SecurityPolicies* this is the DER encoded form of an X.509 v3 *Certificate* as described in Part 6. For ECC *SecurityPolicies* this is an ephemeral key created by the owner of the *KeyCredentials*. Not specified if no encryption is required. If the *securityPolicyUri* is provided this field shall be provided. In 8.4.5 Delete: The *serverNonce* is a random number generated by the GDS. In 8.5.3 Replace: certificateThumbprint: The thumbprint of the *Certificate* used to encrypt the secret. This shall be one of the *ApplicationInstance Certificates* assigned to the *Server*. Not specified if the secret is not encrypted. With: The thumbprint of the *Certificate* used to encrypt the secret. For RSA *SecurityPolicies* this shall be one of the *ApplicationInstance Certificates* assigned to the *Server*. For ECC *SecurityPolicies* this field is not specified. Not specified if the secret is not encrypted.

Topic	GetRejectedList method should be on CertificateGroupType.
Errata Version	1.04.9
Spec Reference	Part 12 7.5.10 CertificateGroupType
Mantis Reference	0004676
Problem Statement	The RejectedList can depend on the Group. Current implementation has one list per Server.

Solution	Add
	7.5.10.1 GetRejectedList
	The *GetRejectedList Method* returns the list of *Certificates* that have been rejected by the *Server* when using the *TrustList* associated with the *CertificateGroup*. It can be used to track activity or allow administrators to move a rejected *Certificate* into the *TrustList*.
	
	No rules are defined for how the *Server* updates this list or how long a *Certificate* is kept in the list. It is recommended that every valid but untrusted *Certificate* be added to the rejected list as long as storage is available. *Servers* should omit older entries from the list returned if the maximum message size is not large enough to allow the entire list to be returned.
	
	This *Method* requires an encrypted channel and that the *Client* provides credentials with administrative rights on the *Server*.
	
	Signature
	
	`GetRejectedList(`
	`[out] ByteString[] certificates`
	`);`
	
	Argument / **Description**
	Certificates / The DER encoded form of the *Certificates* rejected by the *Server*.
	
	Method Result Codes (defined in Call Service)
	
	Result Code / **Description**
	Bad_UserAccessDenied / The current user does not have the rights required.
	Bad_SecurityModeInsufficient / The *SecureChannel* is not encrypted.
	
	Table 21a specifies the *AddressSpace* representation for the *GetRejectedList Method*.
	
	Table 21a – GetRejectedList Method AddressSpace Definition
	
	Attribute / **Value**
	BrowseName / GetRejectedList
	References / **NodeClass** / **BrowseName** / **DataType** / **TypeDefinition** / **ModellingRule**
	HasProperty / Variable / OutputArguments / Argument[] / PropertyType / Mandatory

Topic	TrustListType.OpenWithMasks modeling rule inconsistency with nodeset.
Errata Version	1.04.11
Spec Reference	Part 12 7.5.2 TrustListType
Mantis Reference	0004798
Problem Statement	ModellingRule on OpenWithMasks should be Mandatory
Solution	Replace row in Table 13 with HasComponent | Method | OpenWithMasks | Defined in 7.5.3. | Mandatory

Topic	UnregisterApplication should revoke Certificates.
Errata Version	1.04.11
Spec Reference	Part 12 6.3.8 UnregisterApplication
Mantis Reference	0004262
Problem Statement	*UnregisterApplication* does not have any requirements for what happens to *Certificates* for unregistered *Applications*.
Solution	Add this text to the end of 6.3.8: If an *Application* has *Certificates* issued by the *CertificateManager*, these *Certificates* shall be revoked when this *Method* is called.

Topic	KeyCredentialConfiguration needs to support ECC.
Errata Version	1.04.11
Spec Reference	Part 12 8.5 Information Model for Push Management
Mantis Reference	0004299
Problem Statement	KeyCredentialConfiguration does not support ECC.
Solution	Add 8.5.7 GetEncryptingKey *GetEncryptingKey* is used to request a key that can be used to encrypt a *KeyCredential*. This *Method* requires an encrypted channel and that the *Client* provides credentials with administrative rights on the *Server*. **Signature** **GetEncryptingKey**([in] String credentialId [in] String requestedSecurityPolicyUri [out] ByteString publicKey [out] String revisedSecurityPolicyUri); **Argument** / **Description** credentialId: The unique identifier associated with the *KeyCredential*. requestedSecurityPolicyUri: The *SecurityPolicy* used to encrypt the secret. If not specified the *Server* chooses a suitable default. publicKey: The Public Key used to encrypt the secret. The format depends on the *SecurityPolicyUri*. revisedSecurityPolicyUri: The *SecurityPolicy* used to encrypt the secret. It also specifies the contents of the *publicKey*. This may be different from the *requestedSecurityPolicyUri*. **Method Result Codes (defined in Call Service)** **Result Code** / **Description** Bad_InvalidArgument: The credentialId is not valid. Bad_UserAccessDenied: The current user does not have the rights required. Table 60a specifies the *AddressSpace* representation for the *GetEncryptingKey Method*. **Table 60a – GetEncryptingKey Method AddressSpace Definition** **Attribute** / **Value** BrowseName: 0:GetEncryptingKey **References** / **NodeClass** / **BrowseName** / **DataType** / **TypeDefinition** / **Modelling Rule** 0:HasProperty / Variable / 0:InputArguments / 0:Argument[] / 0:PropertyType / Mandatory 0:HasProperty / Variable / 0:OutputArguments / 0:Argument[] / 0:PropertyType / Mandatory

Topic	StartSigningRequest should allow Applications to access without user credentials.
Errata Version	1.04.11
Spec Reference	Part 12 7.6.3 StartSigningRequest 7.6.4 StartNewKeyPairRequest
Mantis Reference	0004580
Problem Statement	StartNewKeyPairRequest does not make it clear that Applications can invoke the method.
Solution	7.6.4 StartSigningRequest Change Text: This *Method* can be invoked by a configuration tool which has provided user credentials with necessary access permissions. It can also be invoked by the *Application* that owns the private key used to sign the *CertificateRequest* (e.g. the private key shall be the private key used to create the *SecureChannel*). To This *Method* can be invoked by a configuration tool which has provided user credentials with necessary access permissions. It can also be invoked by an *Application* which is requesting an update to a previously issued *Certificate* and that *Certificate* was used to create the *SecureChannel.* 7.6.4 StartNewKeyPairRequest Add Text: This *Method* can be invoked by a configuration tool which has provided user credentials with necessary access permissions. It can also be invoked by an *Application* which is requesting an update to a previously issued *Certificate* and that *Certificate* was used to create the *SecureChannel*.

Topic	CertificateGroupFolderType members should be Organizes
Errata Version	1.04.12
Spec Reference	Part 12 7.5.17 CertificateGroupFolderType
Mantis Reference	0006529
Problem Statement	NodeSet has HasComponent, Document has Organizes. Should change document to match NodeSet.
Solution	Replace Table 28 – CertificateGroupFolderType Definition with:

Attribute	**Value**				
BrowseName	CertificateGroupFolderType				
Namespace	CORE (see 3.3)				
IsAbstract	False				
References	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**
Subtype of the *FolderType* defined in OPC-10000-5.					
HasComponent	Object	DefaultApplicationGroup		CertificateGroupType	Mandatory
HasComponent	Object	DefaultHttpsGroup		CertificateGroupType	Optional
HasComponent	Object	DefaultUserTokenGroup		CertificateGroupType	Optional
Organizes	Object	<AdditionalGroup>		CertificateGroupType	Optional Placeholder

Topic	ApplyChanges Manadatory in NodeSet, Optional in Specification
Errata Version	1.04.12
Spec Reference	Part 12 7.7.3 ServerConfigurationType
Mantis Reference	0006530
Problem Statement	ServerConfigurationType has ApplyChanges Manadatory in NodeSet, Optional in Specification.
Solution	Change row in Table 40 – ServerConfigurationType Definition From :

HasComponent	Method	ApplyChanges	See 7.7.5.	Optional	HasComponent

To:

HasComponent	Method	ApplyChanges	See 7.7.5.	Mandatory	HasComponent

Topic	Need discussion on how to handle existing sessions after a trust list update
Errata Version	1.04.12
Spec Reference	Part 12 7.5.2 TrustListType
Mantis Reference	0006834
Problem Statement	Need discussion on how to handle existing sessions after a trust list update.
Solution	Replace text in 7.8.2.3 CloseAndUpdate with: The *CloseAndUpdate Method* closes the *TrustList* and applies the changes to the *TrustList*. It can only be called if the *TrustList* was opened for writing. If the *Close Method* is called any cached data is discarded and the *TrustList* is not changed. If only part of the *TrustList* is being updated the *Server* creates a new *TrustList* that includes the existing *TrustList* plus any updates and validates the new *TrustList*. The *Server* shall verify that every *Certificate* in the new *TrustList* is valid using the validation process defined in OPC 10000-4. If an invalid *Certificate* is found the *Server* shall return an error and shall not replace the existing *TrustList*. If the *Server* does not support transactions it applies the changes immediately and sets *applyChangesRequired* to FALSE. If the *Server* supports transactions then the *Server* creates a new transaction or continues an existing transaction and sets *applyChangesRequired* to TRUE. If a transaction exists, the *Server* does not update the *TrustList* until *ApplyChanges* is called. Any *Clients* that read the *TrustList* before *ApplyChanges* is called will receive the existing *TrustList* before the transaction started. If no transaction exists then the *Server* sets *applyChangesRequired* to FALSE and updates the *TrustList* before responding to the *Method*. Any *Clients* that read the *TrustList* before this *Method* is called will receive the value that the *TrustList* had before *Open* was called. If errors occur, the new *TrustList* is discarded. When the *TrustList* changes the *Server* shall re-evaluate the *Certificate* associated with any open *Sessions*. *Sessions* with an untrusted or revoked *Certificate* shall be closed.

Topic	Need reference to Part 4 reconnect logic.
Errata Version	1.04.12
Spec Reference	Part 12 7.7.5 ApplyChanges
Mantis Reference	0006835
Problem Statement	Need reference to Part 4 reconnect logic.
Solution	Replace paragraph in 7.7.5 ApplyChanges: If the *Server Certificate* has changed, *Secure Channels* using the old *Certificate* will eventually be interrupted. The only leeway the *Server* has is with the timing. In the best case, the *Server* can close the *TransportConnections* for the affected *Endpoints* and leave any *Subscriptions* intact. This should appear no different than a network interruption from the perspective of the *Client*. The *Client* should be prepared to deal with *Certificate* changes during its reconnect logic. In the worst case, a full shutdown which affects all connected *Clients* will be necessary. In the latter case, the *Server* shall advertise its intent to interrupt connections by setting the *SecondsTillShutdown* and *ShutdownReason Properties* in the *ServerStatus Variable*. With: If the *Server Certificate* has changed, *Secure Channels* using the old *Certificate* will eventually be interrupted. The only leeway the *Server* has is with the timing. In the best case, the *Server* can close the *TransportConnections* for the affected *Endpoints* and leave any *Subscriptions* intact. This should appear no different than a network interruption from the perspective of the *Client*. The *Client* should be prepared to deal with *Certificate* changes during it's reconnect logic as described in OPC 10000-4. In the worst case, a full shutdown which affects all connected *Clients* will be necessary. In the latter case, the *Server* shall advertise its intent to interrupt connections by setting the *SecondsTillShutdown* and *ShutdownReason Properties* in the *ServerStatus Variable*.

Topic	Need discussion on what happens if ApplyChangesRequired=True and ApplyChanges is never called.
Errata Version	1.04.12
Spec Reference	Part 12 7.7.5 ApplyChanges
Mantis Reference	0006837
Problem Statement	Need discussion on what happens if ApplyChangesRequired=True and ApplyChanges is never called.
Solution	Append text to 7.7.5 ApplyChanges: If a *Session* is closed or abandoned then any transaction started when *ApplyChangesRequired*=True is returned by *CloseAndUpdate* or *UpdateCertificate* is closed and all pending changes are discarded.

Topic	Access to a KeyCredentialService is not specified clearly.
Errata Version	1.04.12
Spec Reference	Part 12 8 KeyCredential Management 8.2 Pull Management
Mantis Reference	0006839
Problem Statement	Access to a KeyCredentialService is not specified clearly.
Solution	Change text in 8.2 Pull Management from: *KeyCredentials* can only be requested for *Clients* which are trusted by the *KeyCredentialService*. To: *KeyCredentials* shall only be returned to applications which are authorized by the *KeyCredentialService*.

OPC UA Specification: Part 13 – Aggregates

Topic	Aggregate start, end Delta wording
Errata Version	1.04.12
Spec Reference	Part 13 5.4.3.27 Table 38 Table A.23.2 Start Data Historian1, Historian2, Historian3 Table A.24.2 End Data Historian1, Historian2, Historian3
Mantis Reference	0004062
Problem Statement	The Raw status bit is always set for non-bad StatusCodes for the Start and End aggregates, but it should not be. Delta aggregate has the Raw bit set when it should not have it set.
Solution	5.4.3.27 Table 38 – Delta Aggregate summary The Raw bit description is changed from “Always” to “Not Set”. Replace Table: A.23.2 Start Data

Historian1			
Timestamp	Value	StatusCode	Notes
12:00:10.000	10	Good, Partial	
12:00:20.000	20	Good	
12:00:40.000		Bad	
12:00:50.000	50	Good	
12:01:10.000	70	Uncertain	
12:01:20.000	80	Good, Partial	
12:01:36.000		BadNoData	

Historian2			
Timestamp	Value	StatusCode	Notes
12:00:02.000	10	Good, Partial	
12:00:25.000	20	Good	
12:00:39.000	30	Good	
12:00:48.000	40	Good	
12:01:12.000	60	Good	
12:01:23.000	70	Good, Partial	

12:01:36.000		BadNoData	

Historian3			
Timestamp	Value	StatusCode	Notes
12:00:02.000	10	Good, Partial	
12:00:25.000	20	Good	
12:00:39.000	30	Good	
12:00:48.000	40	Good	
12:01:12.000	60	Good	
12:01:23.000	70	Good, Partial	
12:01:36.000		BadNoData	

With:

Historian1			
Timestamp	Value	StatusCode	Notes
12:00:10.000	10	Good, Raw, Partial	
12:00:20.000	20	Good, Raw	
12:00:40.000		Bad	
12:00:50.000	50	Good, Raw	
12:01:10.000	70	Uncertain, Raw	
12:01:20.000	80	Good, Raw, Partial	
12:01:36.000		BadNoData	

Historian2			
Timestamp	Value	StatusCode	Notes
12:00:02.000	10	Good, Raw, Partial	
12:00:25.000	20	Good, Raw	
12:00:39.000	30	Good, Raw	
12:00:48.000	40	Good, Raw	
12:01:12.000	60	Good, Raw	
12:01:23.000	70	Good, Raw, Partial	
12:01:36.000		BadNoData	

Historian3			
Timestamp	Value	StatusCode	Notes
12:00:02.000	10	Good, Raw, Partial	
12:00:25.000	20	Good, Raw	
12:00:39.000	30	Good, Raw	
12:00:48.000	40	Good, Raw	
12:01:12.000	60	Good, Raw	
12:01:23.000	70	Good, Raw, Partial	
12:01:36.000		BadNoData	

Replace Table: A.24.2 End Data

Historian1			
Timestamp	Value	StatusCode	Notes
12:00:10.000	10	Good, Partial	
12:00:20.000	20	Good	
12:00:40.000		Bad	
12:00:50.000	50	Good	
12:01:10.000	70	Uncertain	
12:01:20.000	80	Good, Partial	
12:01:36.000		BadNoData	

Historian2			
Timestamp	Value	StatusCode	Notes
12:00:02.000	10	Good, Partial	
12:00:25.000	20	Good	

12:00:39.000	30	Good	
12:00:48.000	40	Good	
12:01:12.000	60	Good	
12:01:23.000	70	Good, Partial	
12:01:36.000		BadNoData	

Historian3			
Timestamp	Value	StatusCode	Notes
12:00:02.000	10	Good, Partial	
12:00:25.000	20	Good	
12:00:39.000	30	Good	
12:00:48.000	40	Good	
12:01:12.000	60	Good	
12:01:23.000	70	Good, Partial	
12:01:36.000		BadNoData	

With:

Historian1			
Timestamp	Value	StatusCode	Notes
12:00:10.000	10	Good, Raw, Partial	
12:00:20.000	20	Good, Raw	
12:00:40.000		Bad	
12:00:50.000	50	Good, Raw	
12:01:10.000	70	Uncertain, Raw	
12:01:20.000	80	Good, Raw, Partial	
12:01:36.000		BadNoData	

Historian2			
Timestamp	Value	StatusCode	Notes
12:00:02.000	10	Good, Raw, Partial	
12:00:25.000	20	Good, Raw	
12:00:39.000	30	Good, Raw	
12:00:48.000	40	Good, Raw	
12:01:12.000	60	Good, Raw	
12:01:23.000	70	Good, Raw, Partial	
12:01:36.000		BadNoData	

Historian3			
Timestamp	Value	StatusCode	Notes
12:00:02.000	10	Good, Raw, Partial	
12:00:25.000	20	Good, Raw	
12:00:39.000	30	Good, Raw	
12:00:48.000	40	Good, Raw	
12:01:12.000	60	Good, Raw	
12:01:23.000	70	Good, Raw, Partial	
12:01:36.000		BadNoData	

Topic	Inconsistency for Aggregate Interpolative between description and Sample Historian
Errata Version	1.04.12
Spec Reference	Part 13 Table A.2.2 Interpolative data
Mantis Reference	0005040
Problem Statement	Entries in the Interpolative examples Tables A.2.2 Historian1, Historian2, and Historian3 need to be changed from “Good” to “Good, Raw” status codes when the timestamp matches with the timestamp of the data source.
Solution	Replace Table: A.2.2

Historian1			
Timestamp	Value	StatusCode	Notes
12:00:00.000		BadNoData	
12:00:05.000		BadNoData	
12:00:10.000	10	Good	
12:00:15.000	15	Good, Interpolated	
12:00:20.000	20	Good	
12:00:25.000	25	Good, Interpolated	
12:00:30.000	30	Good	
12:00:35.000	35	UncertainDataSubNormal, Interpolated	
12:00:40.000	40	UncertainDataSubNormal, Interpolated	
12:00:45.000	45	UncertainDataSubNormal, Interpolated	
12:00:50.000	50	Good	
12:00:55.000	55	Good, Interpolated	
12:01:00.000	60	Good	
12:01:05.000	65	UncertainDataSubNormal, Interpolated	
12:01:10.000	70	Uncertain	
12:01:15.000	75	UncertainDataSubNormal, Interpolated	
12:01:20.000	80	Good	
12:01:25.000	85	Good, Interpolated	
12:01:30.000	90	Good	
12:01:35.000	90	UncertainDataSubNormal, Interpolated	

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	102.5	UncertainDataSubNormal, Interpolated	

Historian3			
Timestamp	Value	StatusCode	Notes
12:00:00.000		BadNoData	
12:00:05.000	10	Good, Interpolated	
12:00:10.000	10	Good, Interpolated	
12:00:15.000	10	Good, Interpolated	
12:00:20.000	10	Good, Interpolated	
12:00:25.000	20	Good	
12:00:30.000	25	Good, Interpolated	
12:00:35.000	25	Good, Interpolated	
12:00:40.000	30	Good, Interpolated	
12:00:45.000	30	UncertainDataSubNormal, Interpolated	
12:00:50.000	40	Good, Interpolated	
12:00:55.000	50	Good, Interpolated	
12:01:00.000	50	Good, Interpolated	
12:01:05.000	50	Good, Interpolated	
12:01:10.000	50	Good, Interpolated	
12:01:15.000	60	Good, Interpolated	
12:01:20.000	60	UncertainDataSubNormal, Interpolated	
12:01:25.000	70	Good, Interpolated	
12:01:30.000	90	Good	
12:01:35.000	90	UncertainDataSubNormal, Interpolated	

With:

Historian1			
Timestamp	Value	StatusCode	Notes
12:00:00.000		BadNoData	
12:00:05.000		BadNoData	
12:00:10.000	10	Good, Raw	
12:00:15.000	15	Good, Interpolated	
12:00:20.000	20	Good, Raw	
12:00:25.000	25	Good, Interpolated	
12:00:30.000	30	Good, Raw	
12:00:35.000	35	UncertainDataSubNormal, Interpolated	
12:00:40.000	40	UncertainDataSubNormal, Interpolated	
12:00:45.000	45	UncertainDataSubNormal, Interpolated	
12:00:50.000	50	Good, Raw	
12:00:55.000	55	Good, Interpolated	
12:01:00.000	60	Good, Raw	
12:01:05.000	65	UncertainDataSubNormal, Interpolated	
12:01:10.000	70	Uncertain	
12:01:15.000	75	UncertainDataSubNormal, Interpolated	
12:01:20.000	80	Good, Raw	
12:01:25.000	85	Good, Interpolated	
12:01:30.000	90	Good, Raw	
12:01:35.000	90	UncertainDataSubNormal, Interpolated	

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, Raw	
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, Raw	
12:01:35.000	102.5	UncertainDataSubNormal, Interpolated	

Historian3			
Timestamp	Value	StatusCode	Notes
12:00:00.000		BadNoData	
12:00:05.000	10	Good, Interpolated	
12:00:10.000	10	Good, Interpolated	
12:00:15.000	10	Good, Interpolated	
12:00:20.000	10	Good, Interpolated	
12:00:25.000	20	Good, Raw	
12:00:30.000	25	Good, Interpolated	
12:00:35.000	25	Good, Interpolated	
12:00:40.000	30	Good, Interpolated	
12:00:45.000	30	UncertainDataSubNormal, Interpolated	
12:00:50.000	40	Good, Interpolated	
12:00:55.000	50	Good, Interpolated	
12:01:00.000	50	Good, Interpolated	
12:01:05.000	50	Good, Interpolated	
12:01:10.000	50	Good, Interpolated	
12:01:15.000	60	Good, Interpolated	
12:01:20.000	60	UncertainDataSubNormal, Interpolated	
12:01:25.000	70	Good, Interpolated	
12:01:30.000	90	Good, Raw	
12:01:35.000	90	UncertainDataSubNormal, Interpolated	

Topic	Sample Data missing for DurationInStateZero and DurationInStateNonZero
Errata Version	1.04.12
Spec Reference	Part 13 Table A21.2 Historian Table A.20.2 DurationStateZero Data

	Table A.21.2 DurationStateNonZero Data
Mantis Reference	0005049
Problem Statement	Sample Data missing for DurationInStateZero and DurationInStateNonZero and single value incorrect.
Solution	Replace Table: A.21.2 Historian 4 12:01:20.000 3001 UncertainDataSubNormal, Calculated, Partial With: 12:01:20.000 10000 UncertainDataSubNormal, Calculated, Partial

Add Tables: A.20.2 DurationStateZero

Historian1			
Timestamp	Value	StatusCode	Notes
12:00:00.000	0	UncertainDataSubNormal, Calculated, Partial	
12:00:16.000	0	Good, Calculated	
12:00:32.000	0	UncertainDataSubNormal, Calculated	
12:00:48.000	0	UncertainDataSubNormal, Calculated	
12:01:04.000	0	Good, Calculated	
12:01:20.000	0	Good, Calculated, Partial	
12:01:36.000		BadNoData	

Historian2			
Timestamp	Value	StatusCode	Notes
12:00:00.000	0	UncertainDataSubNormal, Calculated, Partial	
12:00:16.000	0	Good, Calculated	
12:00:32.000	0	UncertainDataSubNormal, Calculated	
12:00:48.000	0	Good, Calculated	
12:01:04.000	0	UncertainDataSubNormal, Calculated	
12:01:20.000	0	UncertainDataSubNormal, Calculated, Partial	
12:01:36.000		BadNoData	

Historian3			
Timestamp	Value	StatusCode	Notes
12:00:00.000	0	Good, Calculated, Partial	
12:00:16.000	0	Good, Calculated	
12:00:32.000	0	Good, Calculated	
12:00:48.000	0	Good, Calculated	
12:01:04.000	0	Good, Calculated	
12:01:20.000	0	Good, Calculated, Partial	
12:01:36.000		BadNoData	

Add Tables: A.21.2 DurationStateNonZero

Historian1			
Timestamp	Value	StatusCode	Notes
12:00:00.000	6000	UncertainDataSubNormal, Calculated, Partial	
12:00:16.000	16000	Good, Calculated	

12:00:32.000	0	UncertainDataSubNormal, Calculated	
12:00:48.000	14000	UncertainDataSubNormal, Calculated	
12:01:04.000	0	Good, Calculated	
12:01:20.000	10001	Good, Calculated, Partial	
12:01:36.000		BadNoData	

Historian2			
Timestamp	Value	StatusCode	Notes
12:00:00.000	14000	UncertainDataSubNormal, Calculated, Partial	
12:00:16.000	16000	Good, Calculated	
12:00:32.000	10000	UncertainDataSubNormal, Calculated	
12:00:48.000	16000	Good, Calculated	
12:01:04.000	13000	UncertainDataSubNormal, Calculated	
12:01:20.000	7001	UncertainDataSubNormal, Calculated, Partial	
12:01:36.000		BadNoData	

Historian3			
Timestamp	Value	StatusCode	Notes
12:00:00.000	14000	Good, Calculated, Partial	
12:00:16.000	16000	Good, Calculated	
12:00:32.000	10000	Good, Calculated	
12:00:48.000	16000	Good, Calculated	
12:01:04.000	13000	Good, Calculated	
12:01:20.000	7001	Good, Calculated, Partial	
12:01:36.000		BadNoData	

Topic	Wrong Data in Sample Historian for Aggregate Delta
Errata Version	1.04.12
Spec Reference	Part 13 Table A.27.2 Delta data
Mantis Reference	0005090
Problem Statement	Whenever the StatusCode is bad the value should be NULL and not have a value of 0 (zero).
Solution	Replace Table: A.27.2 Historian1

Historian1			
Timestamp	Value	StatusCode	Notes
12:00:00.000	0	Good, Calculated, Partial	
12:00:16.000	10	Good, Calculated	
12:00:32.000	0	BadNoData	
12:00:48.000	10	Good, Calculated	
12:01:04.000	0	BadNoData	
12:01:20.000	10	Good, Calculated, Partial	
12:01:36.000		BadNoData	

With:

Historian1			
Timestamp	Value	StatusCode	Notes
12:00:00.000	0	Good, Calculated, Partial	
12:00:16.000	10	Good, Calculated	
12:00:32.000		BadNoData	
12:00:48.000	10	Good, Calculated	

12:01:04.000		BadNoData	
12:01:20.000	10	Good, Calculated, Partial	
12:01:36.000		BadNoData	

Topic	Wrong DataType for StandardDeviationSample and following aggregates
Errata Version	1.04.12
Spec Reference	Part 13 Table 48 – StandardDeviationSample Aggregate summary Table 49 – VarianceSample Aggregate summary Table 50 – StandardDeviationPopulation Aggregate summary Table 51 – VariancePopulation Aggregate summary
Mantis Reference	0005118
Problem Statement	The affected tables have the field Data Type listed as "Status Code" when it should be "Double".
Solution	Replace in tables: 48, 49, 50, and 51 Data Type StatusCode With: Data Type Double

Topic	Wrong Values in Sample Historian_2 and Historian_3 for TimeAverage, in Sample Historian_2 for TimeAverage2
Errata Version	1.04.12
Spec Reference	Part 13 Table A.4.2 TimeAverage data Historian2 and Historian3 Table A.5.2 TimeAverage2 data Historian2
Mantis Reference	0005140
Problem Statement	Values are incorrect due to rounding errors.
Solution	Replace Table: A.4.2 TimeAverage data

Historian2			
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	

12:01:35.000	90	UncertainDataSubNormal, Calculated	

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	

With:

Historian2			
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	27.045	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	96.25	UncertainDataSubNormal, Calculated, Partial	
12:01:35.000		BadNoData	

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	27.045	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		BadNoData	

Replace: A.5.2 TimeAverage2 data

Historian2			
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	27.045	Good, Calculated	
12:00:35.000	29.273	UncertainDataSubNormal, Calculated	
12:00:40.000	30	UncertainDataSubNormal, Calculated	
12:00:45.000	42.500	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	59.800	UncertainDataSubNormal, Calculated	
12:01:15.000	60	UncertainDataSubNormal, Calculated	
12:01:20.000	73.333	UncertainDataSubNormal, Calculated	
12:01:25.000	83.667	Good, Calculated	
12:01:30.000	90	UncertainDataSubNormal, Calculated. Partial	
12:01:35.000		BadNoData	

With:

Historian2			
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	27.045	Good, Calculated	
12:00:35.000	29.273	UncertainDataSubNormal, Calculated	
12:00:40.000		Bad, Calculated	
12:00:45.000	42.500	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	59.800	UncertainDataSubNormal, Calculated	
12:01:15.000	30	Bad, Calculated	
12:01:20.000	73.333	UncertainDataSubNormal, Calculated	
12:01:25.000	83.667	Good, Calculated	
12:01:30.000		Bad, Calculated, Partial	
12:01:35.000		BadNoData	

Topic	Wrong statusCode in Sample Historian_3 for TimeAverage
Errata Version	1.04.12
Spec Reference	Part 13 Table: A.4.2. TimeAverage data Historian3
Mantis Reference	0005141
Problem Statement	The last interval is fully past the last data point so the interval must be Bad.
Solution	Replace Table: A.4.2 Historian3

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	

With:

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	27.045	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		BadNoData	

Topic	Clarification about setting MultipleValues bit in WorstQuality2
Errata Version	1.04.12
Spec Reference	Part 13 5.4.3.35 WorstQuality 5.4.3.36 WorstQuality2
Mantis Reference	0005145
Problem Statement	Wording in description does not match examples. MultipleValues bit should be set whenever there are multiple values.
Solution	Replace: 5.4.3.35 WorstQuality “If multiple values exist with the worst quality but different StatusCodes then the StatusCode of the first value is returned and the MultipleValues bit is set.” With: “If multiple values exist with the worst quality but with different StatusCodes, then the StatusCode of the first value is returned. If multiple values exist with the worst quality, the MultipleValues bit is set.” Replace: 5.4.3.36 WorstQuality2 “If multiple values exist with the worst quality but different StatusCodes then the StatusCode of the first value is returned and the MultipleValues bit is set.” With: “If multiple values exist with the worst quality but with different StatusCodes, then the StatusCode of the first value is returned. If multiple values exist with the worst quality, the MultipleValues bit is set.”

Topic	Example Aggregate data – Historian 2 UseSlopedExtrapolation should be set to true
Errata Version	1.04.12
Spec Reference	Part 13 A.1.2 point 3 Table A.2.2 Interpolative data Hisorian2 Table A.4.2 TimeAverage data Hisorian1
Mantis Reference	0005196
Problem Statement	Example Aggregate data – Historian 2 UseSlopedExtrapolation should be set to true and not false
Solution	Replace Table: A.1.2 point 3 “UseSlopedExtrapolation = False. Therefore SteppedExtrapolation is used at end boundary conditions.” With: “UseSlopedExtrapolation = True. Therefore SteppedExtrapolation is used at end boundary conditions.” Replace Table: A.2.2 Interpolative data Historian2

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	

With:

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, Raw	
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, Raw	
12:01:35.000	102.5	UncertainDataSubNormal, Interpolated	

Replace Table: A.4.2 TimeAverage data Historian1

Historian2			
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	27.045	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		BadNoData	

With:

Historian2			
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	27.045	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	96.25	UncertainDataSubNormal, Calculated, Partial	
12:01:35.000		BadNoData	

Topic	Count data Historian3 Data requires update
Errata Version	1.04.12
Spec Reference	Part 13 Table A5.2 TimeAverage2 Historian3

	Table A7.2 Total2 Historian3 Table A19.2 Count Historian3
Mantis Reference	0005198
Problem Statement	Count and other aggregates require updates because percent good and percent bad are not considered.
Solution	Replace Table: A.5.2 Historian3

Historian3			
Timestamp	Value	StatusCode	Notes
12:00:00.000	10	Good, 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		Bad, Calculated	
12:00:45.000		Bad, 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		Bad, Calculated	
12:01:20.000		Bad, Calculated	
12:01:25.000	78	Good, Calculated	
12:01:30.000	90	Good, Calculated, Partial	
12:01:35.000		BadNoData	

With:

Historian3			
Timestamp	Value	StatusCode	Notes
12:00:00.000	10	Good, 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	40	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	70	UncertainDataSubNormal, Calculated	
12:01:25.000	78	Good, Calculated	
12:01:30.000	90	Good, Calculated, Partial	
12:01:35.000		BadNoData	

Replace Table: A.7.2 Total2 Historian3

Historian3			
Timestamp	Value	StatusCode	Notes
12:00:00.000	30	Good, Calculated, Partial	
12:00:05.000	50	Good, Calculated	
12:00:10.000	50	Good, Calculated	
12:00:15.000	50	Good, Calculated	
12:00:20.000	50	Good, Calculated	
12:00:25.000	110	Good, Calculated	
12:00:30.000	125	Good, Calculated	
12:00:35.000	130	Good, Calculated	
12:00:40.000		Bad, Calculated	
12:00:45.000		Bad, Calculated	
12:00:50.000	230	Good, Calculated	
12:00:55.000	250	Good, Calculated	
12:01:00.000	250	Good, Calculated	
12:01:05.000	250	Good, Calculated	
12:01:10.000	280	Good, Calculated	
12:01:15.000		Bad, Calculated	
12:01:20.000		Bad, Calculated	
12:01:25.000	390	Good, Calculated	
12:01:30.000	0.090	Good, Calculated, Partial	
12:01:35.000		BadNoData	

With:

Historian3			
Timestamp	Value	StatusCode	Notes
12:00:00.000	30	Good, Calculated, Partial	
12:00:05.000	50	Good, Calculated	
12:00:10.000	50	Good, Calculated	
12:00:15.000	50	Good, Calculated	
12:00:20.000	50	Good, Calculated	
12:00:25.000	110	Good, Calculated	
12:00:30.000	125	Good, Calculated	
12:00:35.000	130	Good, Calculated	
12:00:40.000	60	UncertainDataSubNormal, Calculated	
12:00:45.000	80	UncertainDataSubNormal, Calculated	
12:00:50.000	230	Good, Calculated	
12:00:55.000	250	Good, Calculated	
12:01:00.000	250	Good, Calculated	
12:01:05.000	250	Good, Calculated	
12:01:10.000	280	Good, Calculated	
12:01:15.000	120	UncertainDataSubNormal, Calculated	
12:01:20.000	140	UncertainDataSubNormal, Calculated	
12:01:25.000	390	Good, Calculated	
12:01:30.000	0.090	Good, Calculated, Partial	
12:01:35.000		BadNoData	

Replace Table: A.19.2 Count data Historian3

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	

With:

Historian3			
Timestamp	Value	StatusCode	Notes
12:00:00.000	1	Good, Calculated, Partial	
12:00:16.000	2	Good, Calculated	
12:00:32.000	1	Good, Calculated	
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	

Topic	NumberOfTransitions data Historian3 Data requires update
Errata Version	1.04.12
Spec Reference	Part 13 Table A.22.2 Historian3
Mantis Reference	0005199
Problem Statement	PercentGood/PercentBad were not accounted for in the calculation of the aggregate StatusCodes and Values.
Solution	Replace Table: A.22.2 Historian3

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	

With:

Historian3			
Timestamp	Value	StatusCode	Notes
12:00:00.000	1	Good, Calculated, Partial	
12:00:16.000	2	Good, Calculated	
12:00:32.000	1	Good, Calculated	
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	

Topic	Clarification about sample data Historian_3 in Total
Errata Version	1.04.12
Spec Reference	Part 13 Clause 5.4.3.8 Total Table A.6.2 Total data Historian3
Mantis Reference	0005210
Problem Statement	Total aggreagate is calculated using the TimeAverage aggerate. Time average must use SlopedInterpolation but the Time aggregate is incorrectly allowed to used SteppedInterpolation.
Solution	Replace Table: A.22.2 Historian3

Historian3

Timestamp	Value	StatusCode	Notes
12:00:00.000	30	UncertainDataSubNormal, Calculated, Partial	
12:00:05.000	50	Good, Calculated	
12:00:10.000	50	Good, Calculated	
12:00:15.000	50	Good, Calculated	
12:00:20.000	50	Good, Calculated	
12:00:25.000	110	Good, Calculated	
12:00:30.000	125	Good, Calculated	
12:00:35.000	130	Good, Calculated	
12:00:40.000	150	UncertainDataSubNormal, Calculated	
12:00:45.000	170	UncertainDataSubNormal, Calculated	
12:00:50.000	230	Good, Calculated	
12:00:55.000	250	Good, Calculated	
12:01:00.000	250	Good, Calculated	
12:01:05.000	250	Good, Calculated	
12:01:10.000	280	Good, Calculated	
12:01:15.000	300	UncertainDataSubNormal, Calculated	
12:01:20.000	320	UncertainDataSubNormal, Calculated	
12:01:25.000	390	Good, Calculated	
12:01:30.000	450	UncertainDataSubNormal, Calculated, Partial	
12:01:35.000		BadNoData	

With:

Historian3			
Timestamp	Value	StatusCode	Notes
12:00:00.000	53.26	UncertainDataSubNormal, Calculated, Partial	
12:00:05.000	61.955	Good, Calculated	
12:00:10.000	72.825	Good, Calculated	
12:00:15.000	83.695	Good, Calculated	
12:00:20.000	94.565	Good, Calculated	
12:00:25.000	118.41	Good, Calculated	
12:00:30.000	135.225	Good, Calculated	
12:00:35.000	146.92	UncertainDataSubNormal, Calculated	
12:00:40.000	169.445	UncertainDataSubNormal, Calculated	
12:00:45.000	200	UncertainDataSubNormal, Calculated	
12:00:50.000	247.25	Good, Calculated	
12:00:55.000	263.75	Good, Calculated	
12:01:00.000	276.25	Good, Calculated	
12:01:05.000	288.75	Good, Calculated	
12:01:10.000	303.09	Good, Calculated	
12:01:15.000	325	UncertainDataSubNormal, Calculated	
12:01:20.000	352.575	UncertainDataSubNormal, Calculated	
12:01:25.000	418.335	Good, Calculated	
12:01:30.000	450	UncertainDataSubNormal, Calculated, Partial	
12:01:35.000		BadNoData	

Topic	Partial Bit not being set for Interpolated Aggregates TimeAverage and Total
Errata Version	1.04.12
Spec Reference	Part 13 A.4.2 TimeAverage data Historian1 and Historian2 and Historian3 A.6.2 Total data Historian1 and Historian2 and Historian3
Mantis Reference	0005569
Problem Statement	The partial bit in the interval covering the end of the data was not calculated correctly.
Solution	Replace Table Rows: A.4.2 TimeAverage data Historian1 12:01:30.000 | 90 | UncertainDataSubNormal, Calculated | 12:01:35.000 | 90 | UncertainDataSubNormal, Calculated | A.4.2 TimeAverage data Historian2 12:01:30.000 | 90 | UncertainDataSubNormal, Calculated | 12:01:35.000 | 90 | UncertainDataSubNormal, Calculated | A.4.2 TimeAverage data Historian3 12:01:35.000 | 90 | UncertainDataSubNormal, Calculated | With: A.4.2 TimeAverage data Historian1 12:01:30.000 | 90 | UncertainDataSubNormal, Calculated, Partial | 12:01:35.000 | | BadNoData | A.4.2 TimeAverage data Historian2 12:01:30.000 | 96.25 | UncertainDataSubNormal, Calculated, Partial | 12:01:35.000 | | BadNoData | A.4.2 TimeAverage data Historian3 12:01:35.000 | | BadNoData | Replace Table Rows: A.6.2 Total data Historian1 12:01:30.000 | 450 | UncertainDataSubNormal, Calculated | 12:01:35.000 | 450 | UncertainDataSubNormal, Calculated | A.6.2 Total data Historian2 12:01:30.000 | 481.250 | UncertainDataSubNormal, Calculated | 12:01:35.000 | 543.750 | UncertainDataSubNormal, Calculated | A.6.2 Total data Historian3 12:01:30.000 | 450 | UncertainDataSubNormal, Calculated | 12:01:35.000 | 450 | UncertainDataSubNormal, Calculated |

With:

A.6.2 Total data Historian1

12:01:30.000	450	UncertainDataSubNormal, Calculated, Partial	
12:01:35.000		BadNoData	

A.6.2 Total data Historian2

12:01:30.000	481.250	UncertainDataSubNormal, Calculated, Partial	
12:01:35.000		BadNoData	

A.6.2 Total data Historian3

12:01:30.000	450	UncertainDataSubNormal, Calculated, Partial	
12:01:35.000		BadNoData	

Topic	Wording unclear in General section of specification
Errata Version	1.04.12
Spec Reference	Part 13 5.4.3.2.1 General
Mantis Reference	0005579
Problem Statement	The fourth paragraph has the following sentence which is unclear. “The StatusCode of the value is still treated as Uncertain when the StatusCode for the result is calculated.”
Solution	The sentence was removed.

Topic	Wrong value in test data for WorstQuality2/Historian_3
Errata Version	1.04.12
Spec Reference	Part 13 A.34.2 WorstQuality2 data Historian 3
Mantis Reference	0006160
Problem Statement	The calculation for value and status code is incorrect for the rules defined for Historian 3.
Solution	Replace table A.34.2 Historian 3:

Historian3			
Timestamp	Value	StatusCode	Notes
12:00:00.000	BadNoData	Good, Calculated, Partial	
12:00:16.000	Good	Good, Calculated	
12:00:32.000	Bad	Good, Calculated	
12:00:48.000	Good	Good, Calculated	
12:01:04.000	BadNoData	Good, Calculated	
12:01:20.000	BadNoData	Good, Calculated, Partial, MultipleValues	
12:01:36.000		BadNoData	

With:

Historian3			
Timestamp	Value	StatusCode	Notes
12:00:00.000	BadNoData	Good, Calculated, Partial	
12:00:16.000	Good	Good, Calculated	
12:00:32.000	Bad	Good, Calculated	
12:00:48.000	Good	Good, Calculated	
12:01:04.000	UncertainDataSubNormal	Good, Calculated	
12:01:20.000	UncertainDataSubNormal	Good, Calculated, Partial	
12:01:36.000		BadNoData	

Topic	TimeAverage2:Historian1 requires updates as per Get Time Weighted Status
Errata Version	1.04.12
Spec Reference	Part 13 A.5.2 TimeAverage2 data Historian1
Mantis Reference	0006187
Problem Statement	TimeAverage2 Historian1 did not take uncertain regions into account when calculating StatusCodes.
Solution	Replace Table: A.5.2 TimeAverage2 data

Historian1			
Timestamp	Value	StatusCode	Notes
12:00:00.000		BadNoData	
12:00:05.000		BadNoData	
12:00:10.000	12.500	Good, Calculated	
12:00:15.000	17.500	Good, Calculated	
12:00:20.000	22.500	Good, Calculated	
12:00:25.000	27.500	Good, Calculated	
12:00:30.000	30	UncertainDataSubNormal, Calculated	
12:00:35.000	30	UncertainDataSubNormal, Calculated	
12:00:40.000		BadNoData	
12:00:45.000		BadNoData	
12:00:50.000	52.500	Good, Calculated	
12:00:55.000	57.500	Good, Calculated	
12:01:00.000	62.500	UncertainDataSubNormal, Calculated	
12:01:05.000	67.500	UncertainDataSubNormal, Calculated	
12:01:10.000	72.500	UncertainDataSubNormal, Calculated	
12:01:15.000	77.500	UncertainDataSubNormal, Calculated	
12:01:20.000	82.500	Good, Calculated	
12:01:25.000	87.500	Good, Calculated	
12:01:30.000	90	UncertainDataSubNormal, Calculated, Partial	
12:01:35.000		BadNoData	

With:

Historian1			
Timestamp	Value	StatusCode	Notes
12:00:00.000		BadNoData	
12:00:05.000		BadNoData	
12:00:10.000	12.500	Good, Calculated	
12:00:15.000	17.500	Good, Calculated	
12:00:20.000	22.500	Good, Calculated	
12:00:25.000	27.500	Good, Calculated	
12:00:30.000	30	Good, Calculated	
12:00:35.000	30	Good, Calculated	
12:00:40.000		BadNoData	
12:00:45.000		BadNoData	
12:00:50.000	52.500	Good, Calculated	
12:00:55.000	57.500	Good, Calculated	
12:01:00.000	62.500	Good, Calculated	
12:01:05.000	67.500	Good, Calculated	
12:01:10.000	72.500	Good, Calculated	
12:01:15.000	77.500	Good, Calculated	
12:01:20.000	82.500	Good, Calculated	

12:01:25.000	87.500	Good, Calculated	
12:01:30.000	90	Good, Calculated, Partial	
12:01:35.000		BadNoData	

Topic	TimeAverage2:Historian2 requires updates as per Get Time Weighted Status
Errata Version	1.04.12
Spec Reference	Part 13 A.5.2 TimeAverage2 data Historian2
Mantis Reference	0006188
Problem Statement	TimeAverage2 Historian2 did not take uncertain regions into account when calculating StatusCodes.
Solution	Replace Table: A.5.2 TimeAverage2 data

Historian2			
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	29.273	UncertainDataSubNormal, Calculated	
12:00:40.000	30	UncertainDataSubNormal, Calculated	
12:00:45.000	42.500	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	59.800	UncertainDataSubNormal, Calculated	
12:01:15.000	60	UncertainDataSubNormal, Calculated	
12:01:20.000	73.333	UncertainDataSubNormal, Calculated	
12:01:25.000	83.667	Good, Calculated	
12:01:30.000	90	UncertainDataSubNormal, Calculated, Partial	
12:01:35.000		BadNoData	

With:

Historian2			
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	27.045	Good, Calculated	
12:00:35.000	29.273	UncertainDataSubNormal, Calculated	
12:00:40.000		Bad, Calculated	

12:00:45.000	42.500	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	59.800	UncertainDataSubNormal, Calculated	
12:01:15.000	30	Bad, Calculated	
12:01:20.000	73.333	UncertainDataSubNormal, Calculated	
12:01:25.000	83.667	Good, Calculated	
12:01:30.000		Bad, Calculated, Partial	
12:01:35.000		BadNoData	

Topic	Total2:Historian1 requires updates as per Get Time Weighted Status
Errata Version	1.04.12
Spec Reference	Part 13 A.7.2 Total2 data Historian1
Mantis Reference	0006190
Problem Statement	Total2 Historian1 did not take uncertain regions into account when calculating StatusCodes.
Solution	Replace Table: A.7.2 Total2 data

Historian1			
Timestamp	Value	StatusCode	Notes
12:00:00.000		BadNoData	
12:00:05.000		BadNoData	
12:00:10.000	62.500	Good, Calculated	
12:00:15.000	87.500	Good, Calculated	
12:00:20.000	112.500	Good, Calculated	
12:00:25.000	137.500	Good, Calculated	
12:00:30.000	150	UncertainDataSubNormal, Calculated	
12:00:35.000	150	UncertainDataSubNormal, Calculated	
12:00:40.000		BadNoData	
12:00:45.000		BadNoData	
12:00:50.000	262.500	Good, Calculated	
12:00:55.000	287.500	Good, Calculated	
12:01:00.000	312.500	UncertainDataSubNormal, Calculated	
12:01:05.000	337.500	UncertainDataSubNormal, Calculated	
12:01:10.000	362.500	UncertainDataSubNormal, Calculated	
12:01:15.000	387.500	UncertainDataSubNormal, Calculated	
12:01:20.000	412.500	Good, Calculated	
12:01:25.000	437.500	Good, Calculated	
12:01:30.000	0.090	UncertainDataSubNormal, Calculated, Partial	
12:01:35.000		BadNoData	

With:

Historian1			
Timestamp	Value	StatusCode	Notes
12:00:00.000		BadNoData	

12:00:05.000		BadNoData	
12:00:10.000	62.500	Good, Calculated	
12:00:15.000	87.500	Good, Calculated	
12:00:20.000	112.500	Good, Calculated	
12:00:25.000	137.500	Good, Calculated	
12:00:30.000	150	Good, Calculated	
12:00:35.000	150	Good, Calculated	
12:00:40.000		BadNoData	
12:00:45.000		BadNoData	
12:00:50.000	262.500	Good, Calculated	
12:00:55.000	287.500	Good, Calculated	
12:01:00.000	312.500	Good, Calculated	
12:01:05.000	337.500	Good, Calculated	
12:01:10.000	362.500	Good, Calculated	
12:01:15.000	387.500	Good, Calculated	
12:01:20.000	412.500	Good, Calculated	
12:01:25.000	437.500	Good, Calculated	
12:01:30.000	0.090	Good, Calculated, Partial	
12:01:35.000		BadNoData	

Topic	Total2:Historian2 requires updates as per Get Time Weighted Status
Errata Version	1.04.12
Spec Reference	Part 13 A.7.2 Total2 data Historian2
Mantis Reference	0006192
Problem Statement	Total2 Historian2 did not take uncertain regions into account when calculating StatusCodes.
Solution	Replace Table: A.7.2 Total2 data

Historian2			
Timestamp	Value	StatusCode	Notes
12:00:00.000	31.957	UncertainDataSubNormal, Calculated, Partial	
12:00:05.000	61.957	Good, Calculated	
12:00:10.000	72.826	Good, Calculated	
12:00:15.000	83.696	Good, Calculated	
12:00:20.000	94.565	Good, Calculated	
12:00:25.000	118.409	Good, Calculated	
12:00:30.000	135.227	Good, Calculated	
12:00:35.000	146.364	UncertainDataSubNormal, Calculated	
12:00:40.000	60	UncertainDataSubNormal, Calculated	
12:00:45.000	85	UncertainDataSubNormal, Calculated	
12:00:50.000	247.250	Good, Calculated	
12:00:55.000	263.750	Good, Calculated	
12:01:00.000	276.250	Good, Calculated	
12:01:05.000	288.750	Good, Calculated	
12:01:10.000	299	UncertainDataSubNormal, Calculated	
12:01:15.000	120	UncertainDataSubNormal, Calculated	
12:01:20.000	146.667	UncertainDataSubNormal, Calculated	
12:01:25.000	418.333	Good, Calculated	
12:01:30.000	0.090	UncertainDataSubNormal, Calculated, Partial	
12:01:35.000		BadNoData	

With:

Historian2			
Timestamp	Value	StatusCode	Notes
12:00:00.000	31.957	UncertainDataSubNormal, Calculated, Partial	
12:00:05.000	61.957	Good, Calculated	
12:00:10.000	72.826	Good, Calculated	
12:00:15.000	83.696	Good, Calculated	
12:00:20.000	94.565	Good, Calculated	
12:00:25.000	118.409	Good, Calculated	
12:00:30.000	135.227	Good, Calculated	
12:00:35.000	146.364	UncertainDataSubNormal, Calculated	
12:00:40.000		Bad, Calculated	
12:00:45.000	85	UncertainDataSubNormal, Calculated	
12:00:50.000	247.250	Good, Calculated	
12:00:55.000	263.750	Good, Calculated	
12:01:00.000	276.250	Good, Calculated	
12:01:05.000	288.750	Good, Calculated	
12:01:10.000	299	UncertainDataSubNormal, Calculated	
12:01:15.000		Bad, Calculated	
12:01:20.000	146.667	UncertainDataSubNormal, Calculated	
12:01:25.000	418.333	Good, Calculated	
12:01:30.000		Bad, Calculated, Partial	
12:01:35.000		BadNoData	

Topic	Maximum2: Historian1 requires updates as per Get Time Weighted Status
Errata Version	1.04.12
Spec Reference	Part 13 A.14.2 Maximum2 data Historian1
Mantis Reference	0006193
Problem Statement	Minimum2 Historian1 did not take uncertain regions into account when calculating StatusCodes.
Solution	Replace Table: A.14.2 Maximum2 data

Historian1			
Timestamp	Value	StatusCode	Notes
12:00:00.000	16	UncertainDataSubNormal, Interpolated, Partial	
12:00:16.000	30	UncertainDataSubNormal, Calculated, MultipleValues	
12:00:32.000	30	UncertainDataSubNormal, Interpolated	
12:00:48.000	64	UncertainDataSubNormal, Interpolated	
12:01:04.000	80	UncertainDataSubNormal, Calculated	
12:01:20.000	90	UncertainDataSubNormal, Calculated, Partial	
12:01:36.000		BadNoData	

With:

Historian1			
Timestamp	Value	StatusCode	Notes

12:00:00.000	16	UncertainDataSubNormal, Interpolated, Partial	
12:00:16.000	30	Good, Calculated, MultipleValues	
12:00:32.000	30	UncertainDataSubNormal, Interpolated	
12:00:48.000	64	UncertainDataSubNormal, Interpolated	
12:01:04.000	80	Good, Calculated	
12:01:20.000	90	Good, Calculated, Partial	
12:01:36.000		BadNoData	

Topic	MaximumActualTime2:Historian1 requires updates as per Get Time Weighted Status
Errata Version	1.04.12
Spec Reference	Part 13 A.16.2 MaximumActualTime2 data Historian1
Mantis Reference	0006194
Problem Statement	MaximumActualTime2 Historian1 did not take uncertain regions into account when calculating StatusCodes.
Solution	Replace Table: A.16.2 MaximumActualTime2 data

Historian1			
Timestamp	Value	StatusCode	Notes
12:00:15.999	16	UncertainDataSubNormal, Interpolated, Partial	
12:00:30.000	30	UncertainDataSubNormal, MultipleValues	
12:00:32.000	30	UncertainDataSubNormal, Interpolated	
12:01:03.999	64	UncertainDataSubNormal, Interpolated	
12:01:19.999	80	UncertainDataSubNormal, Interpolated	
12:01:30.000	90	UncertainDataSubNormal, Partial	
12:01:36.000		BadNoData	

With:

Historian1			
Timestamp	Value	StatusCode	Notes
12:00:15.999	16	UncertainDataSubNormal, Interpolated, Partial	
12:00:30.000	30	Good, MultipleValues	
12:00:32.000	30	UncertainDataSubNormal, Interpolated	
12:01:03.999	64	UncertainDataSubNormal, Interpolated	
12:01:19.999	80	Good, Interpolated	
12:01:30.000	90	Good, Partial	
12:01:36.000		BadNoData	

Topic	Minimum2:Historian1 requires updates as per Get Time Weighted Status
Errata Version	1.04.12
Spec Reference	Part 13 A.13.2 Minimum2 data Historian1
Mantis Reference	0006202

Problem Statement	Minimum2 Historian1 did not take uncertain regions into account when calculating StatusCodes.
Solution	Replace Table: A.13.2 Minimum2 data

Historian1			
Timestamp	Value	StatusCode	Notes
12:00:00.000	10	UncertainDataSubNormal, Calculated, Partial	
12:00:16.000	16	UncertainDataSubNormal, Interpolated	
12:00:32.000	30	UncertainDataSubNormal, Interpolated	
12:00:48.000	50	UncertainDataSubNormal, Calculated	
12:01:04.000	64	UncertainDataSubNormal, Interpolated	
12:01:20.000	80	UncertainDataSubNormal, Partial	
12:01:36.000		BadNoData	

With:

Historian1			
Timestamp	Value	StatusCode	Notes
12:00:00.000	10	UncertainDataSubNormal, Calculated, Partial	
12:00:16.000	16	Good, Interpolated	
12:00:32.000	30	UncertainDataSubNormal, Interpolated	
12:00:48.000	50	UncertainDataSubNormal, Calculated	
12:01:04.000	64	Good, Interpolated	
12:01:20.000	80	Good, Partial	
12:01:36.000		BadNoData	

Topic	MinimumActualTime2:Historian1 requires updates as per Get Time Weighted Status
Errata Version	1.04.12
Spec Reference	Part 13 A.13.2 MinimumActualTime2 data Historian1
Mantis Reference	0006203
Problem Statement	MinimumActualTime2 Historian1 has the wrong StatusCodes calculated while using the TreatUncertainAsBad flag.
Solution	Replace Table: A.13.2 MinimumActualTime2 data

Historian1			
Timestamp	Value	StatusCode	Notes
12:00:10.000	10	UncertainDataSubNormal, Partial	
12:00:16.000	16	UncertainDataSubNormal, Interpolated	
12:00:32.000	30	UncertainDataSubNormal, Interpolated	
12:00:50.000	50	UncertainDataSubNormal	
12:01:04.000	64	UncertainDataSubNormal, Interpolated	
12:01:20.000	80	UncertainDataSubNormal, Partial	
12:01:36.000		BadNoData	

With:

Historian1

Timestamp	Value	StatusCode	Notes
12:00:10.000	10	UncertainDataSubNormal, Partial	
12:00:16.000	16	Good, Interpolated	
12:00:32.000	30	UncertainDataSubNormal, Interpolated	
12:00:50.000	50	UncertainDataSubNormal	
12:01:04.000	64	Good, Interpolated	
12:01:20.000	80	Good, Partial	
12:01:36.000		BadNoData	

Topic	Range2:Historian1 requires updates as per Get Time Weighted Status
Errata Version	1.04.12
Spec Reference	Part 13 A.17.2 Range2 data Historian1
Mantis Reference	0006204
Problem Statement	Range2 Historian1 did not take uncertain regions into account when calculating StatusCodes.
Solution	Replace Table: A.17.2 Range2 data

Historian1			
Timestamp	Value	StatusCode	Notes
12:00:00.000	6	UncertainDataSubNormal, Calculated, Partial	
12:00:16.000	14	UncertainDataSubNormal, Calculated	
12:00:32.000	0	UncertainDataSubNormal, Calculated	
12:00:48.000	14	UncertainDataSubNormal, Calculated	
12:01:04.000	16	UncertainDataSubNormal, Calculated	
12:01:20.000	10	UncertainDataSubNormal, Calculated, Partial	
12:01:36.000		BadNoData	

With:

Historian1			
Timestamp	Value	StatusCode	Notes
12:00:00.000	6	UncertainDataSubNormal, Calculated, Partial	
12:00:16.000	14	Good, Calculated	
12:00:32.000	0	UncertainDataSubNormal, Calculated	
12:00:48.000	14	UncertainDataSubNormal, Calculated	
12:01:04.000	16	Good, Calculated	
12:01:20.000	10	Good, Calculated, Partial	
12:01:36.000		BadNoData	

Topic	Time Based Status Calculations should respect PercentGood/PercentBad equality check.
Errata Version	1.04.12
Spec Reference	Part 13 Clause 4.2.1.2 Clause 5.4.3.2.1

	Table A.5.2 TimeAverage2 data Table A.7.2 Total2 data
Mantis Reference	0006227
Problem Statement	Calculation of PercentGood/PercentBad wasn't clear on when to calculate the value and when to calculate the StatusCode. The examples for TimeAverage2 and Toal2 were incorrectly calculated.
Solution	In clause 4.2.1.2 add the following sentences to the start of the 5th paragraph. "The following calculations are used to detemine the StatusCode which will be used to calculate the value of the aggregate. Refer to 5.4.3 for details on using these Variables when assigning StatusCodes." In clause 5.4.3.2.1 add the following sentence to the start of the 4th paragraph. "Use the following to determine the StatusCode of a given aggregate." Replace Table: A.5.2 TimeAverage2 data

Historian3			
Timestamp	Value	StatusCode	Notes
12:00:00.000	10	Good, 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		Bad	
12:00:45.000		Bad	
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		Bad	
12:01:20.000		Bad	
12:01:25.000	78	Good, Calculated	
12:01:30.000	90	Good, Calculated, Partial	
12:01:35.000		BadNoData	

With:

Historian3			
Timestamp	Value	StatusCode	Notes
12:00:00.000	10	Good, 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	40	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	70	UncertainDataSubNormal, Calculated	
12:01:25.000	78	Good, Calculated	
12:01:30.000	90	Good, Calculated, Partial	
12:01:35.000		BadNoData	

Replace Table: A.7.2 Total2 data

Historian3			
Timestamp	Value	StatusCode	Notes
12:00:00.000	30	Good, Calculated, Partial	
12:00:05.000	50	Good, Calculated	
12:00:10.000	50	Good, Calculated	
12:00:15.000	50	Good, Calculated	
12:00:20.000	50	Good, Calculated	
12:00:25.000	110	Good, Calculated	
12:00:30.000	125	Good, Calculated	
12:00:35.000	130	Good, Calculated	
12:00:40.000		Bad, Calculated	
12:00:45.000		Bad, Calculated	
12:00:50.000	230	Good, Calculated	
12:00:55.000	250	Good, Calculated	
12:01:00.000	250	Good, Calculated	
12:01:05.000	250	Good, Calculated	
12:01:10.000	280	Good, Calculated	
12:01:15.000		Bad, Calculated	
12:01:20.000		Bad, Calculated	
12:01:25.000	390	Good, Calculated	
12:01:30.000	0.090	Good, Calculated, Partial	
12:01:35.000		BadNoData	

With:

Historian3			
Timestamp	Value	StatusCode	Notes
12:00:00.000	30	Good, Calculated, Partial	
12:00:05.000	50	Good, Calculated	
12:00:10.000	50	Good, Calculated	
12:00:15.000	50	Good, Calculated	
12:00:20.000	50	Good, Calculated	
12:00:25.000	110	Good, Calculated	
12:00:30.000	125	Good, Calculated	
12:00:35.000	130	Good, Calculated	
12:00:40.000	60	UncertainDataSubNormal, Calculated	
12:00:45.000	80	UncertainDataSubNormal, Calculated	
12:00:50.000	230	Good, Calculated	
12:00:55.000	250	Good, Calculated	
12:01:00.000	250	Good, Calculated	
12:01:05.000	250	Good, Calculated	
12:01:10.000	280	Good, Calculated	
12:01:15.000	120	UncertainDataSubNormal, Calculated	
12:01:20.000	140	UncertainDataSubNormal, Calculated	
12:01:25.000	390	Good, Calculated	

	12:01:30.000	0.090	Good, Calculated, Partial	
	12:01:35.000		BadNoData	

OPC UA Specification: Part 14 – PubSub

Topic	Transport of message body with AMQP
Errata Version	1.04.2
Spec Reference	Part 14 7.3.4.8 Message Body
Mantis Reference	0004261
Problem Statement	It is not clear how the message body is included into an AMQP message.
Solution	Add: 7.3.4.8.1 General The message body is encoded in the AMQP bare-message application-data section as an AMQP 'binary' value.

Topic	Handling of large messages in AMQP
Errata Version	1.04.2
Spec Reference	Part 14 7.3.4 AMQP 7.3.4.1 General
Mantis Reference	0004269
Problem Statement	Not clear how do deal with large messages in AMQP.
Solution	Replace: This specification defines two possible message mappings for the AMQP message body, the UADP message mapping defined in 7.2.2 and a JSON message mapping defined in 7.2.3. AMQP *Brokers* have an upper limit on message size. The mechanism for handling *NetworkMessage* that exceed the *Broker* limits depend on the encoding. With: This specification defines two possible message mappings for the AMQP message body, the UADP message mapping defined in 7.2.2 and a JSON message mapping defined in 7.2.3. AMQP Brokers have an upper limit on message size. The limit is defined by the AMQP field max-message-size. The mechanism for handling NetworkMessage that exceed the Broker limits depends on the MessageMapping. For MessageMappings that support chunking, the NetworkMessage shall be broken into multiple chunks. The chunk size plus the AMQP header should not exceed the AMQP max-message-size. For MessageMappings that do not support chunking, the NetworkMessages exceeding the maximum size must be skipped. Diagnostic information for such error scenarios is provided through the Events of the type PubSubTransportLimitsExceedEventType defined in 9.1.12.2 and through the FailedTransmissions counter of the PubSubDiagnosticsWriterGroupType defined in 9.1.11.

Topic	Reference of term that is not known in context of AMQP
Errata Version	1.04.2
Spec Reference	Part 14 7.3.4.8.2 UADP Message Mapping
Mantis Reference	0004270
Problem Statement	Reference of term that is not known in context of AMQP.
Solution	Replace: It is recommended that the MetaDataQueueName as described in 6.4.2.3.6 is configured as a sub-topic of the related QueueName with the name $Metadata. With: It is recommended to create the MetaDataQueueName as described in 6.4.2.3.5 by appending the text “$Metadata” to the related QueueName.

Topic	Wrong DataType for localeIds in WriterGroupDataType
Errata Version	1.04.2
Spec Reference	Part 14 Table 22 – WriterGroupDataType Structure
Mantis Reference	0004292
Problem Statement	DataType for localeIds in WriterGroupDataType must be LocaleId.
Solution	Replace: String[] DataType of localeIds With: LocalId[] DataType of localeIds

Topic	Allow repeated DataSetMessages
Errata Version	1.04.2
Spec Reference	Part 14 6.2.5.2 PublishingInterval 6.2.8.6 MessageReceiveTimeout
Mantis Reference	0004302
Problem Statement	Clarified that DataSetMessages can be repeated with the same SequenceNumber.
Solution	6.2.5.2 PublishingInterval Add: For cyclic *PublishedDataSets* one *DataSet* is produced for one *PublishedDataSet* in a *PublishingInterval*. If no new *DataSet* is available in a *PublishingInterval*, then either the previous *DataSetMessage* is resent or no *DataSetMessage* is sent. 6.2.8.6 MessageReceiveTimeout Replace: If there is no *DataSetMessage* With: If there is no new *DataSetMessage* Add: A *DataSetMessage* is considered new if the sequence number increments. If no sequence number is contained in the *DataSetMessage*, each received *DataSetMessage* is considered new.

Topic	DataSetWriterId field in JSON DataSetMessage
Errata Version	1.04.2
Spec Reference	Part 14 Table 92 – JSON DataSetMessage Definition
Mantis Reference	0004327
Problem Statement	The DataType of the DataSetWriterId in a JSON DataSetMessage is inconsistent with all other DataSetWriterId definitions.
Solution	Table 92 – JSON DataSetMessage Definition Field DataSetWriterId Type Replace: String With: UInt16

Topic	NetworkMessage Header field PayloadHeader
Errata Version	1.04.2
Spec Reference	Part 14 Table 73 – UADP NetworkMessage
Mantis Reference	0004328
Problem Statement	NetworkMessage Header field PayloadHeader refers to wrong bit range.
Solution	Table 73 – UADP NetworkMessage Field PayloadHeader Description Replace: The payload header depends on the UADP NetworkMessage type flags defined in the ExtendedFlags2 bit range 0-3. With: The payload header depends on the UADP NetworkMessage type flags defined in the ExtendedFlags2 bit range 2-4.

Topic	NetworkMessage Header field Payload
Errata Version	1.04.2
Spec Reference	Part 14 Table 73 – UADP NetworkMessage
Mantis Reference	0004330
Problem Statement	NetworkMessage Header field Payload refers to wrong bit range.
Solution	Table 73 – UADP NetworkMessage Field Payload Description Replace: The payload depends on the UADP NetworkMessage Type flags defined in the ExtendedFlags2 bit range 2-5. With: The payload depends on the UADP NetworkMessage Type flags defined in the ExtendedFlags2 bit range 2-4.

Topic	DataSetMessage Header field DataSetMessageSequenceNumber
Errata Version	1.04.2
Spec Reference	Part 14 Table 81 – DataSetMessage Header Structure
Mantis Reference	0004331
Problem Statement	DataSetMessage Header field DataSetMessageSequenceNumber refers to wrong bit.
Solution	Table 81 – DataSetMessage Header Structure Field DataSetMessageSequenceNumber Description Replace: The field shall be omitted if Bit 2 of DataSetFlags1 is false. With: The field shall be omitted if Bit 3 of DataSetFlags1 is false.

Topic	DataSetMessage Header field Timestamp
Errata Version	1.04.2
Spec Reference	Part 14 Table 81 – DataSetMessage Header Structure
Mantis Reference	0004332
Problem Statement	DataSetMessage Header field Timestamp refers to wrong bit.
Solution	Table 81 – DataSetMessage Header Structure Field Timestamp Description Replace: The Timestamp shall be omitted if Bit 3 of DataSetFlags1 is false. With: The Timestamp shall be omitted if Bit 4 of DataSetFlags2 is false.

Topic	DataSetMessage Header field PicoSeconds
Errata Version	1.04.2
Spec Reference	Part 14 Table 81 – DataSetMessage Header Structure
Mantis Reference	0004333
Problem Statement	DataSetMessage Header field PicoSeconds refers to wrong bit.
Solution	Table 81 – DataSetMessage Header Structure Field PicoSeconds Description Replace: The field shall be omitted if Bit 4 of DataSetFlags2 is false. With: The field shall be omitted if Bit 5 of DataSetFlags2 is false.

Topic	Data Key Frame DataSetMessage field DataSetFields
Errata Version	1.04.2
Spec Reference	Part 14 Table 82 – Data Key Frame DataSetMessage Structure
Mantis Reference	0004334
Problem Statement	Data Key Frame DataSetMessage field DataSetFields refers to wrong bits.
Solution	Table 82 – Data Key Frame DataSetMessage Structure Field DataSetFields Description Replace: The default encoding is Variant if bit 0 and 1 are not set. With: The default encoding is Variant if bit 1 and 2 are not set.

Topic	Data Delta Frame DataSetMessage field FieldValue
Errata Version	1.04.2
Spec Reference	Part 14 Table 83 – Data Delta Frame DataSetMessage Structure
Mantis Reference	0004335
Problem Statement	Data Delta Frame DataSetMessage field FieldValue refers to wrong bits.
Solution	Table 83 – Data Delta Frame DataSetMessage Structure Field FieldValue Description Replace: The default encoding is Variant if bit 2 and 3 are not set. With: The default encoding is Variant if bit 1 and 2 are not set.

Topic	NetworkMessage Header field PromotedFields
Errata Version	1.04.2
Spec Reference	Part 14 Table 73 – UADP NetworkMessage
Mantis Reference	0004367
Problem Statement	NetworkMessage Header field PromotedFields refers to wrong bit.
Solution	Table 73 – UADP NetworkMessage Field PromotedFields Description Replace: The PromotedFields shall be omitted if bit 4 of the ExtendedFlags2 is false. With: The PromotedFields shall be omitted if bit 1 of the ExtendedFlags2 is false.

Topic	Clarify SequenceNumber overflow handling
Errata Version	1.04.2
Spec Reference	Part 14 Table 73 – UADP NetworkMessage Table 75 – Layout of the MessageNonce for AES-CTR Table 81 – DataSetMessage Header Structure
Mantis Reference	0004371
Problem Statement	Clarification for resetting sequence number records in Subscriber related to keep alive time if no messages received. This clarification was not contained in all SequenceNumber definitions.

Solution

Replace:
Sequence number detailed description the following tables
Table 73 – UADP NetworkMessage
Table 75 – Layout of the MessageNonce for AES-CTR
Table 81 – DataSetMessage Header Structure

With:
Reference to new chapter 7.2.2

Add new chapter:
7.2.2 SequenceNumber in headers
SequenceNumber fields are defined in different headers of the UADP Message Mapping.
A *SequenceNumber* is a monotonically increasing number assigned to message headers represented by an unsigned integer of width N, which is further specified in Table 73.
Receivers need to be aware of sequence numbers roll over (change from the largest possible value to 0).
To determine whether a received message is newer than the last processed message the following formula shall be used:

(received sequence number -1 - last processed sequence number) modulo 2^N

For the resulting value there is an upper bound and a lower bound depending on the bit width of the sequence number.
Results below the lower bound indicate that the received message is newer than the last processed message and it shall be processed.
Results above the upper bound indicate that the received message is older (or same) than the last processed message and it shall be ignored unless reordering of messages is required.
Other results are invalid, and the message shall be ignored.
The lower bound is given as 2^(N-2).
The upper bound is given as 2^N – 2^(N-2).

Table 73 – Values for different sequence number sizes

DataType	Name	Value	Description
UInt16	Formula	(New-1-Last) modulo 65536	
	Lower bound	16384	2^14
	Upper bound	49152	2^16-2^14
UInt32	Formula	(New-1-Last) modulo 4294967296	
	Lower bound	1073741824	2^30
	Upper bound	3221225472	2^32-2^30

Subscribers shall discard the records they keep for sequence numbers if they do not receive messages for two times the keep alive time to deal with *Publishers* that are out of service and were not able to continue from the last used *SequenceNumber*.

Topic	Clarify content of PubSubConfiguration file
Errata Version	1.04.2
Spec Reference	Part 14 6.2.11 PubSubConfigurationDataType
Mantis Reference	0004396
Problem Statement	The namespaces list is not always null.
Solution	6.2.11 PubSubConfigurationDataType Replace: The values of the UABinaryFileDataType structure are described in Table 50. With: The structure of the UABinaryFileDataType file is described in Table 50. Table 50 – PubSubConfiguration File Content Field namespaces Values Replace: null With: Namespace URIs for namespace indices used in the body. Examples are *NodeIds* contained in *PublishedDataSets*. The OPC UA namespace is skipped.

Topic	Header layout and KeyFrameCount in configuration parameters
Errata Version	1.04.2
Spec Reference	Part 14 6.2.5 WriterGroup Parameters 6.2.8 DataSetReader Parameters
Mantis Reference	0004401, 0004402
Problem Statement	The header layout setting is missing in the configuration parameters for WriterGroup and DataSetReader and the corresponding PubSub configuration model object types. The KeyFrameCount is missing in the DataSetReader parameters and the corresponding PubSub configuration model object type.

Solution

6.2.5 WriterGroup Parameters

Adds:

6.2.5.6 HeaderLayoutUri

The HeaderLayoutUri, with DataType String, defines the selection of a well defined configuration for a subset of the PubSub communication parameters. The affected subset is defined by the header layout.

A null String is defined as no layout selected.

If a layout is selected, all affected parameters shall be set to the values defined for the layout.

Replace Table 22 with:

Name	Type	Description
WriterGroupDataType	Structure	
writerGroupId	UInt16	Defined in 6.2.5.1.
publishingInterval	Duration	Defined in 6.2.5.2.
keepAliveTime	Duration	Defined in 6.2.5.3.
priority	Byte	Defined in 6.2.5.4.
localeIds	LocaleId[]	Defined in 6.2.5.5.
headerLayoutUri	String	Defined in 6.2.5.6.
transportSettings	WriterGroupTransportDataType	Transport mapping specific *WriterGroup* parameters. The abstract base type is defined in 6.2.5.7.2. The concrete subtypes are defined in the sections for transport mapping specific parameters.
messageSettings	WriterGroupMessageDataType	*NetworkMessage* mapping specific *WriterGroup* parameters. The abstract base type is defined in 6.2.5.7.3. The concrete subtypes are defined in the sections for message mapping specific parameters.
dataSetWriters	DataSetWriterDataType[]	The DataSetWriters contained in the *WriterGroup*. The *DataSetWriter* parameters are defined in 6.2.3.

6.2.8 DataSetReader Parameters

Adds:

6.2.8.7 KeyFrameCount

The KeyFrameCount with DataType UInt32 is the multiplier of the PublishingInterval that defines the maximum number of times the PublishingInterval expires before a key frame message, with all field values, is received.

For DataSets that provide cyclic updates, the value shall be greater or equal to 1. For non-cyclic DataSets, like PublishedEvents, that provide event based DataSets, the value shall be 0.

6.2.8.8 HeaderLayoutUri

The HeaderLayoutUri, with DataType String, defines the selection of a well defined configuration for a subset of the PubSub communication parameters. The affected subset is defined by the header layout.

A null String is defined as no layout selected.

If a layout is selected, all affected parameters shall be set to the values defined for the layout.

Replace Table 36 with:

Name	Type	Description
DataSetReaderDataType	Structure	

name	String	The name of the DataSetReader.
enabled	Boolean	The enabled state of the DataSetReader.
publisherId	BaseDataType	Defined in 6.2.8.1.
writerGroupId	UInt16	Defined in 6.2.8.2.
dataSetWriterId	UInt16	Defined in 6.2.8.3.
dataSetMetaData	DataSetMetaDataType	Defined in 6.2.8.4.
dataSetField ContentMask	DataSetFieldContentMask	Defined in 6.2.8.5.
messageReceiveTimeout	Duration	Defined in 6.2.8.6.
keyFrameCount	UInt32	Defined in 6.2.8.7.
headerLayoutUri	String	Defined in 6.2.8.8.
securityMode	MessageSecurityMode	Defined in 6.2.8.9.
securityGroupId	String	Defined in 6.2.8.10.
securityKeyServices	EndpointDescription[]	Defined in 6.2.8.11.
dataSetReaderProperties	KeyValuePair[]	Defined in 6.2.8.12.
transportSettings	DataSetReaderTransportDataType	Transport specific DataSetReader parameters. The abstract base type is defined in 6.2.8.13.2. The concrete subtypes are defined in the sections for transport mapping specific parameters
messageSettings	DataSetReaderMessageDataType	DataSetMessage mapping specific DataSetReader parameters. The abstract base type is defined in 6.2.8.13.3. The concrete subtypes are defined in the sections for message mapping specific parameters.
subscribedDataSet	SubscribedDataSetDataType	The SubscribedDataSet specific parameters. The abstract base type and the concrete subtypes are defined 6.2.9.

Table 116 – WriterGroupType Definition

Add the following row

References	Node Class	BrowseName	DataType	TypeDefinition	Modelling Rule
HasProperty	Variable	HeaderLayoutUri	String	PropertyType	Mandatory

Table 127 – DataSetReaderType Definition

Add the following two rows

References	Node Class	BrowseName	DataType	TypeDefinition	Modelling Rule
HasProperty	Variable	KeyFrameCount	UInt32	PropertyType	Mandatory
HasProperty	Variable	HeaderLayoutUri	String	PropertyType	Mandatory

Topic	Differences to Nodeset.
Errata Version	1.04.3
Spec Reference	Part 14 6.3.2.2.1 DataSetMessageContentMask
Mantis Reference	0004476
Problem Statement	The bits of JsonDataSetMessageContentMask start with 1 but need to start with 0
Solution	6.3.2.2.1 DataSetMessageContentMask Table 3 – JsonDataSetMessageContentMask Values Change bit numbering to start from 0

Value	Bit No.	Description
DataSetWriterId	0	If this flag is set, a DataSetWriterId shall be included in the *DataSetMessage* header.
MetaDataVersion	1	If this flag is set, the *ConfigurationVersion* is included in the *DataSetMessage* header.
SequenceNumber	2	If this flag is set, the DataSetMessageSequenceNumber is included in the DataSetMessage header.
Timestamp	3	If this flag is set, a timestamp shall be included in the *DataSetMessage* header.
Status	4	If this flag is set, an overall status is included in the *DataSetMessage* header.

Topic	Uniqueness of PubSub object names
Errata Version	1.04.3
Spec Reference	Part 14 Table 17 – DataSetWriterDataType Structure Table 20 – PubSubGroupDataType Structure Table 26 – PubSubConnectionDataType Structure Table 36 – DataSetReaderDataType Structure
Mantis Reference	0004519
Problem Statement	It is not clear if names must be unique.
Solution	Add: Table 17 – DataSetWriterDataType Structure Structure field Name The name shall be unique across a WriterGroup. It is recommended to use a human readable name. Table 20 – PubSubGroupDataType Structure Structure field Name The name shall be unique across all writer groups and reader groups of a PubSubConnection. It is recommended to use a human readable name. Table 26 – PubSubConnectionDataType Structure Structure field Name The name shall be unique across a PubSubConfiguration. It is recommended to use a human readable name. Table 36 – DataSetReaderDataType Structure Structure field Name The name shall be unique across a ReaderGroup. It is recommended to use a human readable name.

Topic	Handling of reserved bits and bit ranges in UADP headers
Errata Version	1.04.3
Spec Reference	Part 14 Table 74 – UADP NetworkMessage Table 82 – DataSetMessage Header Structure
Mantis Reference	0004492
Problem Statement	Handling of reserved bits and bit ranges in UADP headers not defined
Solution	Add: Table 74 – UADP NetworkMessage Table 82 – DataSetMessage Header Structure Several places for bit ranges: Reserved values shall not be used by the sender and the receiver shall skip messages when reserved values are received. Several places for reserved bits: Reserved bits shall be set to false by the sender and the receiver shall skip messages where the reserved bits are not false.

Topic	Discovery Message Response flags
Errata Version	1.04.3
Spec Reference	Part 14 7.2.3.4.2 UADP Discovery Response NetworkMessage 7.2.3.4.2.1 General
Mantis Reference	0004500
Problem Statement	Discovery Message Response flags description is incomplete
Solution	7.2.3.4.2 UADP Discovery Response NetworkMessage 7.2.3.4.2.1 General Add: Bit 4 of ExtendedFlags1 shall be true Replace The setting of the flags ensures a known value for the first five fields in the *NetworkMessage* for *Publishers* expected by the *Subscriber*. The actual security settings for the *NetworkMessage* are indicated by the *SecurityHeader*. With The setting of the flags ensures a known value for the first three bytes plus the PublisherId in the *NetworkMessage*, except for the *Chunk* bit 0 in ExtendedFlags2. The actual security settings for the *NetworkMessage* are indicated by the *SecurityHeader*.

Topic	Inconsistency for value rank of PublishingOffset
Errata Version	1.04.3
Spec Reference	Part 14 9.2.1.1 UadpWriterGroupMessageType
Mantis Reference	0004584
Problem Statement	The value rank of PublishingOffset in UadpWriterGroupMessageDataType is 1 and -3 in UadpWriterGroupMessageType
Solution	9.2.1.1 UadpWriterGroupMessageType Replace: The PublishingOffset is defined in 6.3.1.1.6. The ValueRank of the PublishingOffset shall be -3 if the Publisher supports scheduling of multiple NetworkMessages per PublishingInterval. If only a single offset can be configured, the ValueRank shall be -1. Therefore, the Value of the PublishingOffset can be a scalar value or a one-dimensional array value. The default Value is scalar value. With: The *PublishingOffset* is defined in 6.3.1.1.6. In Table 161 – UadpWriterGroupMessageType. Replace: PublishingOffset With: PublishingOffset[]

Topic	Navigation of references between PubSub configuration objects
Errata Version	1.04.3
Spec Reference	Part 14 9.1.3.6 HasPubSubConnection 9.1.5 Connection Model 9.1.6.6 HasDataSetWriter 9.1.6.12 HasDataSetReader
Mantis Reference	0004521, 0004656, 0004675
Problem Statement	The navigation between PubSub configuration objects is easier with special reference types and bidirectional navigation but only a part of the references are specialized PubSub reference types and not all inverse directions are mandatory.

Solution

9.1.3.6 HasPubSubConnection
Add:
Servers shall provide the inverse Reference that relates a PubSubConnection Object back to the PublishSubscribe Object.

9.1.6.6 HasDataSetWriter
Add:
Servers shall provide the inverse Reference that relates a DataSetWriter Object back to a WriterGroupType Object.

9.1.6.12 HasDataSetReader
Add:
Servers shall provide the inverse Reference that relates a DataSetReader Object back to a ReaderGroupType Object.

9.1.5 Connection Model
Add:
9.1.5.7 HasWriterGroup
The HasWriterGroup ReferenceType is a concrete ReferenceType that can be used directly. It is a subtype of the HasComponent ReferenceType.
The SourceNode of References of this type shall be an instance of the PubSubConnectionType defined in 9.1.5.2.
The TargetNode of this ReferenceType shall be an instance of the WriterGroupType defined in 9.1.6.3.
Servers shall provide the inverse Reference that relates a WriterGroup Object back to a PubSubConnectionType Object.
The representation of the HasWriterGroup ReferenceType in the AddressSpace is specified in Table 115.

Table 115 – HasWriterGroup ReferenceType

Attributes	Value		
BrowseName	HasWriterGroup		
InverseName	IsWriterGroupOf		
Symmetric	False		
IsAbstract	False		
References	**NodeClass**	**BrowseName**	**Comment**
Subtype of HasComponent defined in Part 5.			

9.1.5.8 HasReaderGroup
The HasReaderGroup ReferenceType is a concrete ReferenceType that can be used directly. It is a subtype of the HasComponent ReferenceType.
The SourceNode of References of this type shall be an instance of the PubSubConnectionType defined in 9.1.5.2.
The TargetNode of this ReferenceType shall be an instance of the ReaderGroupType defined in 9.1.6.6.
Servers shall provide the inverse Reference that relates a ReaderGroup Object back to a PubSubConnectionType Object.
The representation of the HasReaderGroup ReferenceType in the AddressSpace is specified in Table 116.

Table 116 – HasReaderGroup ReferenceType

Attributes	Value		
BrowseName	HasReaderGroup		
InverseName	IsReaderGroupOf		
Symmetric	False		
IsAbstract	False		
References	**NodeClass**	**BrowseName**	**Comment**
Subtype of HasComponent defined in Part 5.			

Topic	UADP Message Security parameters
Errata Version	1.04.3
Spec Reference	Part 14 7.2.3.2.3 UADP Message Security
Mantis Reference	0004537
Problem Statement	UADP Message Security parameters require clarification
Solution	7.2.3.2.3 UADP Message Security 7.2.3.2.3.1 General Replace: The algorithm and nonce length used of the UADP NetworkMessage security depend on the selected SecurityPolicy. They are defined by SymmetricPubSubEncryptionAlgorithm and SymmetricPubSubNonceLength. The keys used to encrypt and sign messages are returned from the GetSecurityKeys method (see 8.4). This Method returns a sequence of random data with a length that depends on the SecurityPolicyUri, which is also returned by the Method. The layout of the random data is defined in Table 75. With: The security algorithms used and the length of the KeyNonce in the UADP NetworkMessage depend on the selected SecurityPolicy. The algorithms are defined by SymmetricEncryptionAlgorithm and SymmetricSignatureAlgorithm in Part 7. The nonce length is part of the SymmetricEncryptionAlgorithm. The keys used to encrypt and sign messages are returned from the GetSecurityKeys method (see 8.4). This Method returns a sequence of key data with a length that depends on the SecurityPolicyUri, which is also returned by the Method. The layout of the key data is defined in Table 75.

Topic	RawData Field Encoding
Errata Version	1.04.3 This errata is superseded by 1.04.6.
Spec Reference	Part 14 7.2.3.3 DataSetMessage
Mantis Reference	0004592, 0004649
Problem Statement	A fixed offset for fields in DataSetMessages with RawData encoding is expected but is not possible if fields with dynamic size are included.

Solution	7.2.3.3 DataSetMessage Table 82 – DataSetMessage Header Structure DataSetFlags1 Bit range 1-2 01 RawData Field Encoding <u>Replace</u> description with reference to new chapter 7.2.3.3.9 RawData Field Encoding <u>Add new chapter</u>: 7.2.3.3.9 RawData Field Encoding The encoding of the *DataSetMessage* fields is handled like a *Structure DataType* where the *DataSet* fields are handled like *Structure* fields and fields with *Structure DataType* are handled like nested structures. All restrictions for the encoding of *Structure DataTypes* also apply to the *RawData Field Encoding*. A *DataSet* field is encoded in the *DataType* and *ValueRank* specified in the *DataSetMetaData* for the *DataSet*. The following special handling shall be applied • If the *DataType* of a *DataSet* field or a *Structure* field is *String* or *ByteString* and the actual size is smaller than the maximum possible size indicated by the dimensions, the field shall be padded with bytes with value zero. • If the *ValueRank* is *OneDimension* (1) or n>1 and the actual size is smaller than the maximum possible size indicated by the dimensions, the field shall be padded with bytes with value zero. The following restrictions apply to the *RawData* field encoding • Fields shall have dimensions defined if the *DataType* is *String* or *ByteString* or if it is an array. This includes *Structure* fields with such *DataTypes* or *ValueRank*. • *DataSet* fields and *Structure* fields shall have a concrete valueRank with values -1 or n > 0. • *DataSet* fields and *Structure* fields shall not have the *builtInType NodeId*, *ExpandedNodeId*, *QualifiedName*, *LocalizedText*, *XmlElement* or *DataValue*. • *Structure* fields shall not have the *builtInType ExtensionObject* or *Variant*. The *DataSetMessage* valid bit 0 in *DataSetFlags1* shall be set to false if the fields do not fulfil these requirements at the time the *DataSetMessage* is created.

Topic	Repetition of FieldIndex in a Data Delta Frame
Errata Version	1.04.3
Spec Reference	Part 14 7.2.3.3.6 Data Delta Frame DataSetMessage
Mantis Reference	0004498
Problem Statement	Handling of FieldIndex repetition requires clarification.
Solution	7.2.3.3.6 Data Delta Frame DataSetMessage Table 84 – Data Delta Frame DataSetMessage Structure Description FieldIndex Add: A Publisher shall use an index only once in a DataSetMessage.

Topic	Padding in DataSetMessages
Errata Version	1.04.3
Spec Reference	Part 14 7.2.3.3.5 Data Key Frame DataSetMessage 7.2.3.3.6 Data Delta Frame DataSetMessage 7.2.3.3.7 Event DataSetMessage
Mantis Reference	0004551
Problem Statement	The optional padding in DataSetMessages is not part of the message definitions.
Solution	7.2.3.3.5 Data Key Frame DataSetMessage Table 83 – Data Key Frame DataSetMessage Structure 7.2.3.3.6 Data Delta Frame DataSetMessage Table 84 – Data Delta Frame DataSetMessage Structure 7.2.3.3.7 Event DataSetMessage Table 85 – Event DataSetMessage Structure Add the following field to the end of the tables: Padding | Byte [*] | Optional padding added if the encoded DataSetMessage is smaller than the *ConfiguredSize*. The *DataSetMessage* is padded with bytes with value zero.

Topic	Reassembling of chunked NetworkMessage
Errata Version	1.04.3
Spec Reference	Part 14 7.2.3.2.4 UADP Chunk NetworkMessage
Mantis Reference	0004495
Problem Statement	It is not clear how to reassemble chunked NetworkMessages with out of order delivery of chunks
Solution	7.2.3.2.4 UADP Chunk NetworkMessage Table 79 – Chunked NetworkMessage Payload Fields ChunkOffset Replace Description with: The byte offset position of the chunk in the complete NetworkMessage payload. The last chunk is detected if ChunkOffset plus the size of the current chunk equals TotalSize. All chunks, except for the last one shall have the same size. The size of all chunks other than the last one can be used to calculate the number of expected chunks. The reassembled NetworkMessage payload can be processed after all chunks are received. Depending on the transport protocol mapping, the chunks may be received out of order and the last chunk may be received before all other chunks are received.

Topic	Handling of receiverIndexRange with single element selection
Errata Version	1.04.3
Spec Reference	Part 14 6.2.9.2.3 FieldTargetDataType
Mantis Reference	0004776
Problem Statement	If the data selected with receiverIndexRange is one array element, it is not possible to write this value to a variable with ValueRank scalar
Solution	6.2.9.2.3 FieldTargetDataType Table 44 – FieldTargetDataType Structure receiverIndexRange Add to Description: If the resulting array size is one and the target node *ValueRank* is scalar, the value shall be applied as scalar value.

Topic	Error handling for difference between UADP headers and configuration
Errata Version	1.04.4
Spec Reference	Part 14 7.2.2 UADP message mapping
Mantis Reference	0004506, 0004514, 0004570, 0004572
Problem Statement	There is no definition for error handling if UADP headers and message content is different than expected for the configuration
Solution	Add following chapter: **7.2.2.2 Error handling** The PubSub communication parameters defined in Clause 6 provide the settings for mapping information from the *Publisher* into *DataSetMessages*, settings to send them in *NetworkMessages* to the *Subscribers* and settings to process the *DataSetMessages* on the *Subscriber* side. The error handling for the status codes in *DataSetMessage* headers and *DataSetMessage* fields is defined in 6.2.10 and 6.2.3.2. This handling of information flows and status codes assumes that the configuration between *Publisher* and *Subscriber* is in sync. In several combinations of settings for the *DataSetMessages* and *NetworkMessages*, a *Subscriber* is able to process received messages without further knowledge of the *Publisher* side configuration. But most Subscribers need at least the *DataSetMetaData* to be able to process the received *DataSetMessages*. The *Publisher* side configuration implies two types of contracts necessary for the Subscriber to process messages. The one type of contract is the *DataSetMetaData* describing the content of a *DataSetMessage*. The other type of contract provides the communication settings like the *DataSetWriterId* or offsets inside *NetworkMessages*. Both type of contracts provide version information that can be included into the *DataSetMessages* and *NetworkMessages*. Several settings in the contracts have corresponding flags or version fields in the messages and a *Subscriber* can detect mismatches between the contract and the received messages. The error handling depends on the *Subscriber* applications. *Subscribers* that are configured to process certain *DataSetMessages* often work with a known contract and they will typically drop messages that do not comply with the contract. At the same time they will try to get an update of the contract and will try to adjust its own settings to the updated contract if this is possible. Some changes may need manual reconfiguration of the Subscriber. But Subscribers may also work without a known contract or may accept some differences between the contract and the actual message layout without dropping messages. One exception is the security configuration. A *Subscriber* shall drop all messages where the configured *SecurityMode* has a lower number than the received *SecurityMode*. E.g. if the *Subscriber* is configured for *SecurityMode* SIGN_2 it shall drop messages with NONE_1. A *Subscriber* may process messages with a higher *SecurityMode* e.g. it is allowed to process messages with *SecurityMode* SIGN_2 if it is configured for NONE_1.

Topic	Definition of message field and message header status mapping
Errata Version	1.04.4
Spec Reference	Part 14 Table 16 – DataSetMessage field representation options
Mantis Reference	0004583
Problem Statement	Not all necessary combinations of message field and message header status mappings are defined
Solution	Replace Table 16 – DataSetMessage field representation options With the following table

DataSet Publisher		**Field**	**DataSetMessage**			**DataSet Subscriber**	
Value	**Status(1)**		**Value**	**Value Status(1)**	**Header Status**	**Value**	**Status(1)**
Value 1	Good_*	Variant	Value 1	N/A	Good	Value 1	Good
Value 1	Uncertain_*		Value 1 Uncertain_* (2)		Good	Value 1 (2)	Uncertain_* (2)
Null	Bad_*		Bad_* (3)		Good	Null	Bad_* (3)
Value 1	Good_*	DataValue	Value 1	Good_*	Good	Value 1	Good_*
Value 1	Uncertain_*		Value 1	Uncertain_*	Good	Value 1	Uncertain_*
Null	Bad_*		Null	Bad_*	Good	Null	Bad_*
Value 1	Good_*	RawData	Value 1	N/A	Good	Value 1	Good
Value 1	Uncertain_*		Value 1(4)		Uncertain (4)	Value 1	Uncertain
Null	Bad_*		Default value for DataType (5)		Uncertain_SubNormal (5)	Default value for DataType	Uncertain_SubNormal
All fields Bad_*			Default value for DataType (5)		Bad (5)	Null	Bad
The header status is set to a bad code in a fatal error situation.					Bad_*	Null	Bad_*

Note 1: If no specific *StatusCode* is used, the grouping into severity *Good*, *Uncertain* or *Bad* is used. In this case, the resulting *Status* matches the input *Status*.
Note 2: If the status is uncertain in variant encoding, the value and the status are encoded as *DataValue*.
Note 3: A bad status is transferred instead of a value for the variant encoding.
Note 4: If the status for one or more fields is uncertain in raw filed encoding, the header status shall be set to *Uncertain*.
Note 5: If the worst status for some fields is bad, the header status shall be set to *Uncertain_SubNormal*. If the status for all fields is bad, the header status shall be set to *Bad*. The value in message is set to the default value for the *DataType*.

Topic	Default discovery address for UDP protocol mapping
Errata Version	1.04.4
Spec Reference	Part 14 7.3.2 OPC UA UDP
Mantis Reference	0003668
Problem Statement	There is no known IPv4 multicast address for discovery
Solution	7.3.2 OPC UA UDP Add the following paragraph: The IANA registered IPv4 multicast address for discovery is 224.0.2.14. It shall only be used for OPC UA discovery purposes. The default port for discovery is 4840. Therefore the default *DiscoveryAddress* has the following form: opc.udp://224.0.2.14

Topic	Content of discovery response messages and relation to request
Errata Version	1.04.4
Spec Reference	Part 14 Table 87 – Publisher Information Request Message structure
Mantis Reference	0004797
Problem Statement	The result of a discovery request may not fit in one discovery response. It is not clear how to handle such situations.
Solution	Table 87 – Publisher Information Request Message structure Field DataSetWriterIds Add the following paragraphs to the description: For DataSetMetaData requests, the Publisher sends one discovery response NetworkMessage for each requested DataSetWriterId. For DataSetWriter configuration requests, the DataSetWriters that belong to one WriterGroup are sent together in one DataSetWriter configuration message. If more than one WriterGroup is affected, this results in a DataSetWriter configuration message per WriterGroup.

Topic	Aggregation of discovery response messages
Errata Version	1.04.4
Spec Reference	Part 14 7.2.2.6.1.2 Traffic reduction
Mantis Reference	0004569
Problem Statement	Discovery response message aggregation unclear.
Solution	7.2.2.6.1.2 Traffic reduction Replace: Discovery requests for different DataSetWriters in one WriterGroup shall be aggregated into one discovery response. With: If the Publisher receives discovery requests for different DataSetWriters in one WriterGroup, the Publisher shall send one aggregated discovery response.

Topic	Definition of null PublisherId
Errata Version	1.04.4
Spec Reference	Part 14 6.2.6.1 PublisherId
Mantis Reference	0004776
Problem Statement	The meaning of null PublisherIds is not defined
Solution	6.2.6.1 PublisherId Add the following sentence: A zero *UInteger* and a Null or empty *String* are invalid *PublisherIds*.

Topic	DataType of GroupVersion in UadpWriterGroupMessageDataType
Errata Version	1.04.4
Spec Reference	Part 14 Table 54 – UadpWriterGroupMessageDataType structure
Mantis Reference	0004848
Problem Statement	The DataType of the GroupVersion in UadpWriterGroupMessageDataType is different from the DataType in UadpWriterGroupMessageType
Solution	Table 54 – UadpWriterGroupMessageDataType structure Field groupVersion Replace DataType UInt32 with DataType VersionTime

Topic	Range of PicoSeconds in UADP protocol
Errata Version	1.04.4
Spec Reference	Part 14 Table 74 – UADP NetworkMessage Table 82 – DataSetMessage header structure
Mantis Reference	0004621
Problem Statement	The valid range of PicoSeconds fields in the UADP protocol is not defined
Solution	Table 74 – UADP NetworkMessage Field PicoSeconds Table 82 – DataSetMessage header structure Field PicoSeconds Add the following text: The PicoSeconds field stores the difference between a high-resolution timestamp with a resolution of 10 picoseconds and the Timestamp field value which only has a 100 ns resolution. The PicoSeconds field shall contain values less than 10 000. The decoder shall treat values greater than or equal to 10 000 as the value ‘9999’.

Topic	Handling of bad status for target variables that do not support writing of status
Errata Version	1.04.5
Spec Reference	Part 14 6.2.10 Information flow and status handling
Mantis Reference	0005035
Problem Statement	Handling of bad status for target variables that do not support writing of status is not defined
Solution	6.2.10 Information flow and status handling Add following text If one of the target Variables in the SubscribedDataSet does not allow writing of the StatusCode and the OverrideValueHandling is set to Disabled_0, the DataSetReader shall indicate the configuration error by setting the DataSetReader state to Error_3. In all other configurations of OverrideValueHandling when the target Variable does not allow writing of the StatusCode, only the Value is transferred to the target Variable.

Topic	Timestamps in AMQP standard property mapping cannot be configured
Errata Version	1.04.5
Spec Reference	Part 14 Table 98 - OPC UA AMQP standard header QualifiedName Name mappings
Mantis Reference	0005342
Problem Statement	The configuration of AMQP standard properties cannot handle dynamic timestamps. An alternative
Solution	Table 98 - OPC UA AMQP standard header QualifiedName Name mappings Replace table with the following table

AMQP standard property name	OPC UA DataType	AMQP data type	Note
to	String	*	
user-id	ByteString	binary	
reply-to	String	string	
correlation-id	ByteString	*	
absolute-expiry-time	Duration	timestamp	The absolute-expiry-time is calculated by adding the message-absolute-expiry-time (*Duration*) from the *DataSetWriterProperties* to the current time of the *DataSetMessage* creation.
group-id	String	string	
reply-to-group-id	String	string	
creation-time	Boolean	timestamp	The creation-time is set to the current time of the *DataSetMessage* creation if the message-creation-time (*Boolean*) in the *DataSetWriterProperties* is True, or else if the value is False or if the property is not configured, the AMQP property is not set.
content-encoding	String	symbol	

Topic	Definition for RawData field encoding
Errata Version	1.04.6
Spec Reference	Part 14 7.2.2.5.9 RawData field encoding
Mantis Reference	0005484, 0005471, 0005472, 0005476, 0005473
Problem Statement	The list of restrictions was not complete and not all special cases were covered.

Solution	7.2.2.5.9 RawData field encoding Replace complete chapter with the following text: The encoding of the *DataSetMessage* fields is handled like a *Structure DataType* where the *DataSet* fields are handled like *Structure* fields and fields with *Structure DataType* are handled like nested structures. All restrictions for the encoding of *Structure DataTypes* also apply to the *RawData Field Encoding*. A *DataSet* field is encoded in the *DataType* and *ValueRank* specified in the *DataSetMetaData* for the *DataSet*. The following special handling shall be applied to ensure a fixed offset of the fields in the *DataSetMessage*. • If the *DataType* of a *DataSet* field or a *Structure* field is *String* or *ByteString* and the actual size is smaller than the maximum possible size indicated by the dimensions, the field shall be padded with bytes with value zero. • If the *ValueRank* is *OneDimension* (1) or n>1 and the actual size of a dimension is smaller than the maximum possible size indicated by the dimensions, the field shall be padded with bytes with value zero for each dimension. • If the *DataSet* field or *Structure* field is a *Structure* with optional fields, the *EncodingMask* is encoded followed by all fields. Any optional field that is not present is encoded as padding with bytes with value zero. The size of the padding equals to the size needed to encode the field if it were present. • If the *DataSet* field or *Structure* field is a *Union*, the encoding of the selected field is padded with bytes with value zero to the size of the longest *Union* field, when encoded using the rules in this chapter. The case when no field is selected is treated as if there was an encoded field whose encoded size is zero. The following restrictions apply to the *RawData* field encoding. • Fields shall have *maxStringLength* defined in the *FieldMetaData* if the *DataType* is *String* or *ByteString*. Fields shall have *arrayDimensions* defined in the *FieldMetaData* if valueRank has a value of n > 0. This includes *Structure* fields with such *DataTypes* or *ValueRank*. • *DataSet* fields and *Structure* fields shall not have an abstract *DataType*. • *DataSet* fields and *Structure* fields shall have a concrete valueRank with values -1 or n > 0. • *DataSet* fields and *Structure* fields shall not have the *builtInType NodeId*, *ExpandedNodeId*, *QualifiedName*, *LocalizedText*, *XmlElement, DiagnosticInfo* or *DataValue*. • *RawField* encoding shall only be applied to *Data Key Frame DataSetMessages*. The *DataSetMessage* valid bit 0 in *DataSetFlags1* shall be set to false if the fields do not fulfil these requirements at the time the *DataSetMessage* is created.

Topic	Definition of QueueName for Broker DataSetReader
Errata Version	1.04.6
Spec Reference	Part 14 6.4.2.6.1 QueueName
Mantis Reference	0005439
Problem Statement	Improper limitation for QueueName that applies to DataSetWriter but not to DataSetReader.
Solution	6.4.2.6.1 QueueName Remove the following sentence from the definition This parameter is only valid if the *NetworkMessages* from the *WriterGroup* contain only one *DataSetMessage*.

Topic	Inconsistency for optional/mandatory of DataSetWriterId in JSON DataSetMessages
Errata Version	1.04.6
Spec Reference	Part 14 7.2.3.3 DataSetMessage DataSetWriterId
Mantis Reference	0005368, 0005417
Problem Statement	The DataSetWriterId is mandatory in 7.2.3.3 DataSetMessage but can be switched off with the JsonDataSetMessageContentMask
Solution	7.2.3.3 DataSetMessage DataSetWriterId Changed The value is mandatory To The value is optional

Topic	Missing state in PubSubState
Errata Version	1.04.6
Spec Reference	Part 14 6.2.1 PubSubState state machine
Mantis Reference	0005382
Problem Statement	There is a state missing in PubSubState that covers the case where the transition to Enabled takes some time.
Solution	6.2.1 PubSubState state machine Table 1 – PubSubState values Added Value 'PreOperational_4' with the following description The PubSub component is enabled but currently in the process to execute the steps necessary to enter the Operational_2 state. Table 3 – PubSubState state machine Replace table with the following table:

Source State	Target State	Trigger Description
Disabled_0	Paused_1	The component was successfully enabled but the parent component is in the state Disabled_0 or Paused_1.
Disabled_0	PreOperational_4	The component was successfully enabled.
Paused_1	Disabled_0	The component was successfully disabled.
Paused_1	PreOperational_4	The state of the parent component changed to Operational_2.
PreOperational_4	Operational_2	The component completed the steps necessary to enter the *Operational_2* state. If the PubSub component is a *DataSetReader*, the state shall change to *Operational_2* after the first key frame or event *DataSetMessage* was received.
PreOperational_4	Disabled_0	The component was successfully disabled.
PreOperational_4	Paused_1	The state of the parent component changed to Disabled_0 or Paused_1.
PreOperational_4	Error_3	There is a pending error situation for the related *PubSub* component.
Operational_2	Disabled_0	The component was successfully disabled.
Operational_2	Paused_1	The state of the parent component changed to Disabled_0 or Paused_1.
Operational_2	Error_3	There is a pending error situation for the related *PubSub* component.
Error_3	Disabled_0	The component was successfully disabled.
Error_3	Paused_1	The state of the parent component changed to Disabled_0 or Paused_1.
Error_3	Operational_2	The error situation was resolved for the related *PubSub* component.
Error_3	PreOperational_4	The error situation was resolved for the related *PubSub* component.

Topic	Add MLD to provide IGMP like functionality for IPv6
Errata Version	1.04.10
Spec Reference	Part 14 7.3.2 OPC UA UDP
Mantis Reference	0004192
Problem Statement	IGMP is only used for IPv4. There is no corresponding definition for IPv6.
Solution	7.3.2 OPC UA UDP Replace IGMP related definition with the following text: It is recommended to use switches with IGMP and MLD support to limit the distribution of multicast traffic to the interested participants. Note: The Internet Group Management Protocol (IGMP) is a standard protocol used by hosts to report their IP multicast group memberships for IPv4 and needs to be implemented by any host that wishes to receive IP multicast datagrams. IGMP messages are used by multicast routers to learn which multicast groups have members on their attached networks. IGMP messages are also used by switches capable of supporting “IGMP snooping” whereby the switch listens to IGMP messages and only sends the multicast *NetworkMessages* to ports that have joined the multicast group. The corresponding protocol for IPv6 is the Multicast Listener Discovery (MLD). There are different versions of IGMP and MLD: - IGMP V1 is defined in IETF RFC 1112. - IGMP V2 is defined in IETF RFC 2236. - IGMP V3 is defined in IETF RFC 3376. - IGMP V3 and MLD V2 are defined in IETF RFC 4604. IETF RFC 2236 and IETF RFC 3376 discuss host and router requirements for interoperation with older IGMP versions. Since OPC UA devices make extensive use of IP multicast for UDP transport, consistent IGMP and MLD usage by OPC UA devices is essential in order to create well-functioning OPC UA *Application* networks. OPC UA *Applications* shall issue an IGMP membership report message (V1, V2 or V3 as appropriate) for IPv4 or a MLD membership report message for IPv6 when opening a UDP connection on which they will receive multicast *NetworkMessages*.

Topic	MaxStringLength use for LocalizedText
Errata Version	1.04.10
Spec Reference	Part 14 6.2.2.1.3 FieldMetaData
Mantis Reference	0006516
Problem Statement	MaxStringLength does not apply to LocalizedText
Solution	6.2.2.1.3 FieldMetaData Table 5 – FieldMetaData Structure maxStringLength Replace description with: If the dataType field is a String, LocalizedText or ByteString then this field specifies the maximum supported length. If the maximum is unknown the value shall be 0. If the dataType field is not a String, LocalizedText or ByteString the value shall be 0. If the valueRank is greater than 0 this field applies to each element of the array.

Topic	GetSecurityKeys parameter TimeToNextKey
Errata Version	1.04.10
Spec Reference	Part 14 8.4 GetSecurityKeys Method
Mantis Reference	0006327
Problem Statement	GetSecurityKeys parameter TimeToNextKey needs clarification

Solution	8.4 GetSecurityKeys Method Replace Publishers using a central SKS shall call GetSecurityKeys at a period of half the KeyLifetime. With Publishers using a central SKS shall call GetSecurityKeys always with StartingTokenId set to 0 and shall call the Method at a period of half the KeyLifetime. They can still request more than one key to bridge longer unavailability time of the SKS. Subscribers should use a StartingTokenId of 0 the first time they call GetSecurityKeys. Subsequent call to request older or future keys can use specific StartingTokenIds. Description Argument TimeToNextKey: Replace The time, in milliseconds, before the CurrentKey is expected to expire. If a Publisher uses this Method to get the keys from a SKS, the TimeToNextKey and KeyLifetime are used to calculate the time the Publisher shall use the next key. The TimeToNextKey defines the time when to switch from CurrentKey to FutureKeys and the KeyLifetime defines when to switch from one future key to the next future key. For a Subscriber the TimeToNextKey and KeyLifetime are used to calculate the time the Subscriber expects that the Publishers use the next key. Due to network latency, out of order delivery and the use of keys for several Publishers, a Subscriber needs to expect some overlap time where NetworkMessages are received that are using the previous or the next key. TimeToNextKey and KeyLifetime are also used to calculate the time until Publisher and Subscriber shall fetch new keys. With The time, in milliseconds, before the current key is expected to expire. The current SecurityTokenId equals the FirstTokenId and the current key is the first one in the returned Keys if the passed StartingTokenId is 0. Therefore the Method shall be called with StartingTokenId set to 0 if there is no previous knowledge about the current key. If a Publisher uses this Method to get the keys from a SKS, the TimeToNextKey and KeyLifetime are used to calculate the time the Publisher shall use the next key. The TimeToNextKey defines the time when to switch from the current key to the next key and the KeyLifetime defines when to switch from one future key to the next future key. For a Subscriber the TimeToNextKey and KeyLifetime are used to calculate the time the Subscriber expects that the Publishers use the next key. Due to network latency, out of order delivery and the use of keys for several Publishers, a Subscriber needs to expect some overlap time where NetworkMessages are received that are using the previous or the next key.

	TimeToNextKey and KeyLifetime are also used to calculate the time until Publisher and Subscriber shall fetch new keys.

Topic	Identification of SKS in SetSecurityKeys
Errata Version	1.04.10
Spec Reference	Part 14 9.1.3.3 SetSecurityKeys
Mantis Reference	0006329
Problem Statement	SecurityGroupId is only unique in a SKS but SetSecurityKeys does not have a the SKS information
Solution	9.1.3.3 SetSecurityKeys Added following text: The OPC UA Client calling this Method shall be the SKS application with the ApplicationUri that matches the ApplicationUri in the SecurityKeyServices parameter of the WriterGroup, ReaderGroup or DataSetReader objects using the SecurityGroupId.

Topic	AddSecurityGroup method parameter error handling
Errata Version	1.04.10
Spec Reference	Part 14 8.8 AddSecurityGroup Method
Mantis Reference	0006656
Problem Statement	Error handling for AddSecurityGroup method parameters not completely defined
Solution	8.8 AddSecurityGroup Method Added following additional descriptions to existing method argument descriptions: KeyLifetime If 0 is passed in, the SKS sets the default KeyLifetime. If the requested value exceeds the limits defined by the SKS, the value is adjusted by the SKS. The caller should get the revised value by reading the KeyLifetime of the created SecurityGroup. SecurityPolicyUri If a null or empty String is passed in, the SKS sets the default SecurityPolicyUri. If the SecurityPolicyUri is not known to the SKS, Bad_InvalidArgument shall be returned. MaxFutureKeyCount If 0 is passed in, the SKS sets the default MaxFutureKeyCount. If the requested value exceeds the limits defined by the SKS, the value is adjusted by the SKS. The caller should get the revised value by reading the MaxFutureKeyCount of the created SecurityGroup. MaxPastKeyCount If the requested value exceeds the limits defined by the SKS, the value is adjusted by the SKS. The caller should get the revised value by reading the MaxPastKeyCount of the created SecurityGroup. Add to Method Result codes: ResultCode = Bad_InvalidArgument Description = The SecurityPolicyUri is not supported by the SKS.

Topic	MessageType in JSON DataSetMessage
Errata Version	1.04.10
Spec Reference	Part 14 Table 62 – JsonDataSetMessageContentMask Values Table 92 – JSON DataSetMessage Definition
Mantis Reference	0005372
Problem Statement	The flag to enable the MessageType field is missing in the JsonDataSetMessageContentMask.
Solution	Table 62 – JsonDataSetMessageContentMask Values Add row to table: Value = MessageType Bit No. = 5 Description = If this flag is set, the message type is included in the DataSetMessage header. Table 92 – JSON DataSetMessage Definition Add row to table: Name = MessageType Type = String Description = Possible values are “ua-keyframe”, “ua-deltaframe”, “ua-event” and “ua-keepalive”. The presence of the value depends on the setting in the JsonDataSetMessageContentMask.

Topic	AES-CTR BlockCounter starting value
Errata Version	1.04.10
Spec Reference	Part 14 Table 76 – Layout of the counter block for UADP message security
Mantis Reference	0006852
Problem Statement	The AES-CTR BlockCounter starting value is defined as 1 in the RFC but was defined as 0 in Part 14.
Solution	Table 76 – Layout of the counter block for UADP message security BlockCounter Replace The counter starts with 0 at the first block. With The counter starts with 1 at the first block.

Topic	JsonNetworkMessageContentMask
Errata Version	1.04.12
Spec Reference	Part 14 6.3.2.1.1 NetworkMessageContentMask 7.2.3.2 NetworkMessage
Mantis Reference	0008715
Problem Statement	Additional configuration parameter needed for JsonNetworkMessageContentMask
Solution	6.3.2.1.1 NetworkMessageContentMask Add the following mask value to table: Value bit 6: WriterGroupName Description: The WriterGroup name is included in the *NetworkMessages*. 7.2.3.2 NetworkMessage Add the following message field to the table: Name: WriterGroupName Type: String Description: The name of the WriterGroup which created the NetworkMessage. The value is optional. The presence of the value depends on the setting in the JsonNetworkMessageContentMask.

Topic	DataSetMessageContentMask
Errata Version	1.04.12
Spec Reference	Part 14 6.3.2.2.1 DataSetMessageContentMask
Mantis Reference	0007424, 0008714, 0008715, 0008922,
Problem Statement	Additional configuration parameters needed for DataSetMessageContentMask
Solution	6.3.2.2.1 DataSetMessageContentMask Add the following mask values to table: Value bit 5: MessageType Description: If this flag is set, the message type is included in the *DataSetMessage* header. Value bit 6: DataSetWriterName Description: If this flag is set, a DataSetWriterName shall be included in the *DataSetMessage* header. Value bit 7: ReversibleFieldEncoding Description: If this flag is set, the DataSetMessage fields are encoded in the reversible JSON encoding and in the non-reversible JSON encoding otherwise. Value bit 8: PublisherId Description: The PublisherId is included in the DataSetMessages. This bit

	shall be false if the NetworkMessageHeader is active. Value bit 9: WriterGroupName Description: The WriterGroup name is included in the DataSetMessages. If the WriterGroup name is included in the NetworkMessage header, it shall not be included in the DataSetMessages. Value bit 10: MinorVersion Description: If this flag is set, the MinorVersion field of the ConfigurationVersion is included in the DataSetMessage header. 7.2.3.3 DataSetMessage Add the following message fields to the table: Name: MessageType Type: String Description: Possible values are “ua-keyframe”, “ua-deltaframe”, “ua-event” and “ua-keepalive”. The presence of the value depends on the setting in the JsonDataSetMessageContentMask. Name: DataSetWriterName Type: String Description: The name of the DataSetWriter which created the DataSetMessage. The presence of the value depends on the setting in the JsonDataSetMessageContentMask. Name: PublisherId Type: String Description: A unique identifier for the Publisher. It identifies the source of the message. The presence of the value depends on the setting in the JsonDataSetMessageContentMask. The source is the PublisherId on a PubSubConnection. If the PublisherId is a UInteger, the UInteger value is converted to a String without leading zeros. The value shall be omitted if the NetworkMessage header is present. Name: WriterGroupName Type: String Description: The name of the WriterGroup which created the DataSetMessage. The presence of the value depends on the setting in the JsonDataSetMessageContentMask. The value shall be omitted if the WriterGroupName is contained in the NetworkMessage header. Name: MinorVersion Type: VersionTime Description: The minor version of the DataSetMetaData which describes the contents of the Payload. The presence of the value depends on the setting in the JsonDataSetMessageContentMask. The value shall be omitted if the *MetaDataVersion* is contained in the *DataSetMessage* header.

Topic	Start of periodic execution calculation
Errata Version	1.04.12
Spec Reference	Part 14 6.3.1.1.1 Relationship of Timing parameters
Mantis Reference	0008925
Problem Statement	Formular for start of periodic execution calculation is broken
Solution	6.3.1.1.1 Relationship of Timing parameters Replace Start of periodic execution = current time + PublishingInterval – (current time % PublishingInterval) With Start of periodic execution = current time + PublishingInterval – (current time MODULO PublishingInterval)

Topic	Encoding of JSON event message fields
Errata Version	1.04.12
Spec Reference	Part 14 Table 159 – JSON DataSetMessage definition
Mantis Reference	0007425
Problem Statement	UADP defines restrictions for encoding of event message fields wich is missing for JSON event messages
Solution	Table 159 – JSON DataSetMessage definition Payload Add For MessageType “ua-event”, only Variant or RawData encoding shall be allowed. If bits for DataValue encoding are set, the Variant encoding shall be used.

Topic	FieldMetaData MaxStringLength
Errata Version	1.04.12
Spec Reference	Part 14 6.2.3.2.3 FieldMetaData
Mantis Reference	0008051
Problem Statement	Use of MaxStringLength field in FieldMetaData requires clarification
Solution	6.2.3.2.3 FieldMetaData Table 7 – FieldMetaData structure MaxStringLength Replace If the dataType field is a String, LocalizedText or ByteString then this field specifies the maximum supported length. With If the dataType field is a String, LocalizedText (the text field) or ByteString then this field specifies the maximum supported length of the data in number of bytes.

OPC UA Specification: Part 15 – Safety

Topic	Need safety protocol support.
Errata Version	1.04.7
Spec Reference	Part 15 - Safety
Mantis Reference	2410
Problem Statement	Need safety protocol support.
Solution	Publish: Part 15. https://opcfoundation.org/developer-tools/specifications-unified-architecture/

OPC UA Specification: Part 17 – Alias Names

Topic	The ability to support alternate name for nodes in a system.
Errata Version	1.04.6
Spec Reference	Part 17 – Alias Names
Mantis Reference	5556
Problem Statement	It has been requested to provide a manner of configuring and exposing an alternate well-defined name for any Node in the system. These names may reflect tags that can be subscribed for or Object that can be browsed. They may reflect item like published topics or any other name. they should be searchable to discover a name.
Solution	Publish: Part 17. https://opcfoundation.org/developer-tools/specifications-unified-architecture/

Topic	Mismatch of reference types used for AliasNameCategoryType
Errata Version	1.04.8
Spec Reference	Part 17 – Alias Names Table 2 – AliasNameCategoryType Defintion
Mantis Reference	5768
Problem Statement	In the ObjectType definition of AliasNameCategoryType, the nested SubAliasNameCategories are defined with HasComponent, but In the instances, Organizes is used between Aliases and TagVariables / Topics.
Solution	Change the reference type from HasComponent to Organizes for the <SubAliasNameCategories> Object

Topic	ModelChangeEvent definitions need clarifications
Errata Version	1.04.12
Spec Reference	Part 17 – Alias Names Multiple Parts Table 2 – AliasNameCategoryType Defintion Text below table Table 9 - Aliases definition Text below table Annex A.4 Annex B.2
Mantis Reference	7669
Problem Statement	ModelChange event do not work well enough for AliasNames – mulitiple issue with them – they do not scale for GDS or larger aggregating server, the NodeVersion property that is required for ModelChange does not help for Aliasname and double increase the size required for an AliasName..

Solution

Added a LastChange variable to AliasNameCategoryType and remove reference to ModelChangeEvents.

Add the following row to Table 2

HasProperty	Variable	LastChange	VersionTime	PropertyType	Optional

Add the following text to the end of the text below the table:

"LastChange is the most recent time for any of the following activities:

- The last time an *AliasName* was added to or deleted from the *AliasNameCategory*,
- The last time an *AliasNameCategory* was added or deleted,
- The last time the referenced *Nodes* of an *AliasName* in the *AliasNameCategory* changed.

For *AliasNameCategoryType* instances that are nested, the value of *LastChange* shall always be the latest *VersionTime* of all *Organized AliasName* and *AliasNameCategories*.

The *LastChange* shall be persisted. A *Client* that detects a *LastChange* that is older than what it has cached, shall clear all cached *AliasNameCategories* and related *AliasNames*."

Add the following row to table 9

HasProperty	Variable	LastChange	VersionTime	PropertyType

Replace the following paragraph below table 9

"If the *Server* supports *ModelChange*Events for *AliasNames*, this *Node* shall contain the *NodeVersion Property*. If a *Server* supports the *NodeVersion Property* for this *Node*, then it shall support the *NodeVersion Property* for all instances of *AliasNameType and AliasNameCategoryType.* A *Client* can check this required *Node* to determine if the *Server* supports *ModelChangeEvents* for *AliasNames"*

With the following paragraph

"The *LastChange Property*, which is optional in the *AliasNameCatagoryType*, is mandatory for the *Aliases* instance. This property shall reflect the last time any changes to AliasName occurred (see 6.3)."

From annex A.4

Delete the following sentence from the first paragraph:
"The standalone *AliasNames Server* might have a subscription to each *Server* monitoring the status of the *Server* and monitoring for *ModelChangeEvents* that occur on the source *Servers."*

Delete the following paragraph from annex B.2, below figure B.9

"The GDS shall subscribe for *ModelChangeEvents* form the *Server*, to ensure that any changes to the *AliasNames* would be reflected in the GDS. [note: not all *Servers* will support *ModelChangeEvents*, but AliasName Aggregating *Server* shall support ModelChangeEvents]. When the GDS receives a *ModelChangeEvent* it shall update as needed and generate a *ModelChangeEvent* for its *Clients* - in essence the original event is forwarded to listening *Clients*."

Topic	AliasName and namespaces
Errata Version	1.04.12
Spec Reference	Part 17 – Alias Names Section 6.2 AliasNameType ObjectType Definition
Mantis Reference	7671
Problem Statement	The AliasName feature is supposed to be independent of namespace, but this is nto clear in the specification.
Solution	Add the following sentence to the end of the paragraph under table 1 "A client shall always ignore the namespace associated with an AliasName for comparison with other *AliasNames*."

OPC UA Specification: Part 19 – Dictionary Reference

Topic	PA-DIM companion spec needs subtypes of MultiStateValueDiscreteType that have dictionary references.
Errata Version	1.04.5
Spec Reference	Part 19 – Dictionary Reference
Mantis Reference	5380
Problem Statement	PA-DIM companion spec needs subtypes of MultiStateValueDiscreteType that have dictionary references.
Solution	Publish: Part 19. https://opcfoundation.org/developer-tools/specifications-unified-architecture/

Topic	MultiStateValueDiscreteType is inconsistent with NodeSet
Errata Version	1.04.8
Spec Reference	Clause 7.1 Table 6, definition of EnumDictionaryEntries and definition of ValueAsDictionaryEntries
Mantis Reference	0005862
Problem Statement	ValueRank mismatch between MultiStateDictionaryEntryDiscreteBaseType and its base type.
Solution	Change *ValueRank* from Scalar to Any as the NodeSet defined it. Added description explaining that the ValueAsDictionaryEntries component an only be used with scalar DataTypes.

OPC UA Specification: Part 22 – Base Network Model

Topic	Define common information model for networks
Errata Version	1.04.10
Spec Reference	Part 22 – Base Network Model
Mantis Reference	7004
Problem Statement	There is the need to define a base model for networks so that everybody modelling networks can use this common model as base.
Solution	Publish: Part 22. https://opcfoundation.org/developer-tools/specifications-unified-architecture/

OPC UA Specification: Amendment 1 – AnalogItem Types

Topic	Enhances the UA DataAccess information model defined in Part 8 by adding additional sub-types of DataItemType
Errata Version	1.04.3
Spec Reference	Amendment 1 AnalogItem Types
Mantis Reference	0004229, 004230
Problem Statement	Need VariableType with mandatory EngineeringUnits.
Solution	Publish: Amendment 1. https://opcfoundation.org/developer-tools/specifications-unified-architecture/errata-and-amendments/

OPC UA Specification: Amendment 2 – ChoiceStates and Guards

Topic	Enhances the UA information model for state machines defined in Annex B of Part 5 to include the concepts of ChoiceStates and Guards.
Errata Version	1.04.3
Spec Reference	Amendment 2 ChoiceStates and Guards
Mantis Reference	4277
Problem Statement	Need advanced state machine features
Solution	Publish: Amendment 2. https://opcfoundation.org/developer-tools/specifications-unified-architecture/errata-and-amendments/

OPC UA Specification: Amendment 3 – Method Metadata

Topic	Enhances the UA information model for methods defined in Parts 3, 4 and 5 to support optional arguments and additional metadata such as engineering units.
Errata Version	1.04.3
Spec Reference	Amendment 3 Method Metadata
Mantis Reference	2765; 3792; 4175
Problem Statement	Need support for additional information about method arguments.
Solution	Publish: Amendment 3. https://opcfoundation.org/developer-tools/specifications-unified-architecture/errata-and-amendments/

Topic	Description of the Call Service inputArguments Parameter is incorrect
Errata Version	1.04.3
Spec Reference	Amendment 3 Part 4 Table 65 Call Service Parameters
Mantis Reference	0004467
Problem Statement	The last paragraph of the parameter inputArguments[] description is incorrect.
Solution	Replace: If a Method has optional input arguments, these optional input arguments are indicated by the Method's DefaultInputValues Property. In this case fewer arguments than the total number of input arguments defined may be passed by the Client. The InputArguments and DefaultInputValues Property are defined in Part 3. With: Fewer arguments than the total number of input arguments defined may be passed by the Client when optional input arguments are defined. A Method may define input arguments as optional by including HasOptionalInputArgumentDescription references to argument Metadata. The InputArguments Property and the HasOptionalInputArgumentDescription ReferenceType are defined in Part 3.

Topic	Description of Result Code is incorrect
Errata Version	1.04.3
Spec Reference	Amendment 3 Part 4 Table 67 Call Operation Level Result Codes
Mantis Reference	0004320
Problem Statement	The description of Result Code Bad_ArgumentsMissing was not changed to reflect the option input arguments introduced by Amendment 3
Solution	Replace: The Client did not specify all of the input arguments for the Method. With: The Client did not specify all of the non-optional input arguments for the Method.

OPC UA Specification: Amendment 4 – ECC

Topic	Request for support of Elliptic Curve Cryptography (ECC)
Errata Version	1.04.8
Spec Reference	Amendment 4 ECC
Mantis Reference	0003819, 0004650, 0004299
Problem Statement	Request for support of Elliptic Curve Cryptography (ECC)
Solution	Publish: Amendment 4. https://opcfoundation.org/developer-tools/specifications-unified-architecture/errata-and-amendments/

OPC UA Specification: Amendment 5 – Dictionary Reference

Topic	Describes the basic infrastructure to reference from an OPC UA Information Model to external dictionaries like IEC Common Data Dictionary or eCl@ss.
Errata Version	1.04.3
Spec Reference	Amendment 5 Dictionary Reference
Mantis Reference	0004853
Problem Statement	Need a basic infrastructure to reference from an OPC UA Information Model to external dictionaries like IEC Common Data Dictionary or eCl@ss.
Solution	Publish: Amendment 5. https://opcfoundation.org/developer-tools/specifications-unified-architecture/errata-and-amendments/

OPC UA Specification: Amendment 6 – UADP Header Layouts

Topic	Need a fixed message for PubSub that can be used for optimizing communication.
Errata Version	1.04.3
Spec Reference	Amendment 6 UADP Header Layouts
Mantis Reference	0004530
Problem Statement	Need a fixed message for PubSub that can be used for optimizing communication.
Solution	Publish: Amendment 6. https://opcfoundation.org/developer-tools/specifications-unified-architecture/errata-and-amendments/

Topic	Nonce length in message headers for periodic data with fixed layout
Errata Version	1.04.10
Spec Reference	Amendment 6 UADP Header Layouts Table C.3: UADP NetworkMessage header layout with integrity (signing) Table C.4: UADP NetworkMessage header layout with integrity and confidentiality
Mantis Reference	0006634
Problem Statement	The nonce length is defined as 4 Byte length but it has a length of 8 Byte.
Solution	Table C.3: UADP NetworkMessage header layout with integrity (signing) Table C.4: UADP NetworkMessage header layout with integrity and confidentiality Replace 4 Byte length for NonceLength and MessageNonce with 8 Byte.

Topic	PublisherId type in message headers for periodic data with fixed layout
Errata Version	1.04.10
Spec Reference	Amendment 6 UADP Header Layouts C.2 Message headers for periodic data with fixed layout
Mantis Reference	0006991
Problem Statement	The PublisherId is limited to UInt16. Different use cases for periodic data with fixed layout require a UInt64 PublisherId.
Solution	C.2 Message headers for periodic data with fixed layout Add UInt64 as valid PublisherId option.

OPC UA Specification: Amendment 7 – Interfaces and AddIns

Topic	Enhances the UA type model to support interfaces and object aggregation.
Errata Version	1.04.3
Spec Reference	Amendment 7 Interfaces and AddIns
Mantis Reference	4049
Problem Statement	Need support for (limited) multiple inheritance via interfaces and object aggregation.
Solution	Publish: Amendment 7. https://opcfoundation.org/developer-tools/specifications-unified-architecture/errata-and-amendments/

Topic	DefaultInstanceBrowseName is required for AddIns
Errata Version	1.04.11
Spec Reference	Amendment 7 Interfaces and AddIns
Mantis Reference	7304
Problem Statement	The Addin ObjectType shall include a default BrowseName.
Solution	In 4.9.3 replace the first 3 paragraphs with: *AddIns* associate a feature or feature-set, represented by an *ObjectType* to the *Node* (an *Object* or *ObjectType*) they are applied to. The *Interface* model is different than the *AddIn* model in that it is based on composition. An *AddIn* is applied to a *Node* by adding a *Reference* to the *AddIn* instance. There are no restrictions for *AddIn ObjectTypes* and there is no special supertype for *AddIns*. To identify instances as an *AddIn*, the *HasAddIn Reference* or a subtype shall be used. The *AddIn ObjectType* shall include the definition of a default *BrowseName* using the *DefaultInstanceBrowseName Property*. Instances of such an *AddIn* should use this default *BrowseName*. If an *AddIn* is instantiated multiple times in the same parent, only one instance can have the default *BrowseName*.

OPC UA Specification: Amendment 10 – Currency

Topic	Need a definition for Currency (it is not covered by units)
Errata Version	1.04.5
Spec Reference	Amendment 10 Currency
Mantis Reference	4526
Problem Statement	Need a definition for Currency (it is not covered by units)
Solution	Publish: Amendment 10. https://opcfoundation.org/developer-tools/specifications-unified-architecture/errata-and-amendments/

OPC UA Specification: Amendment 11 – Spatial Types

Topic	Need additional UA types to support multidimensional coordinate systems
Errata Version	1.04.3
Spec Reference	Amendment 11 Spatial Types
Mantis Reference	0002919
Problem Statement	Need additional UA types to support multidimensional coordinate systems
Solution	Publish: Amendment 11. https://opcfoundation.org/developer-tools/specifications-unified-architecture/errata-and-amendments/

OPC UA Specification: Amendment 13 – Ordered List

Topic	Request for standardized ordered list of objects
Errata Version	1.04.7
Spec Reference	Amendment 13 Ordered List
Mantis Reference	0005565
Problem Statement	Request for standardized ordered list of objects
Solution	Publish: Amendment 13. https://opcfoundation.org/developer-tools/specifications-unified-architecture/errata-and-amendments/

OPC UA Specification: NodeSets and Generated Files

Topic	Nodeset contains undefined modelling rule MandatoryShared
Errata Version	1.04.3
Spec Reference	TargetVersion="1.04.3" TargetPublicationDate="2019-09-09T00:00:00Z"
Mantis Reference	0004749
Problem Statement	NodeSet contains undefined modelling rule MandatoryShared.
Solution	Delete: Reference to undefined Object ‘MandatoryShared’.

Topic	NodeSet has wrong ValueRank for UadpWriterGroupMessageType PublishingOffset.
Errata Version	1.04.3
Spec Reference	TargetVersion="1.04.3" TargetPublicationDate="2019-09-09T00:00:00Z"
Mantis Reference	0004762
Problem Statement	NodeSet has wrong ValueRank for UadpWriterGroupMessageType PublishingOffset.
Solution	Update: Set ValueRank="1" for UadpWriterGroupMessageType/PublishingOffset.

Topic	Commas in CSV fields are not enclosed in double quotes.
Errata Version	1.04.3
Spec Reference	TargetVersion="1.04.3" TargetPublicationDate="2019-09-09T00:00:00Z"
Mantis Reference	0004788
Problem Statement	Commas in CSV fields are not enclosed in double quotes.
Solution	Update: Add double quotes around every description.

Topic	UadpWriterGroupMessageDataType.GroupVersion has wrong DataType.
Errata Version	1.04.4
Spec Reference	TargetVersion="1.04.4" TargetPublicationDate="2019-11-20T00:00:00Z"
Mantis Reference	0005027
Problem Statement	GroupVersion has wrong DataType.
Solution	In UadpWriterGroupMessageDataType Change: GroupVersion DataType from UInt32 to VersionTime.

Topic	HasWriterGroup and HasReaderGroup are missing.
Errata Version	1.04.4
Spec Reference	TargetVersion="1.04.4" TargetPublicationDate="2019-11-20T00:00:00Z"
Mantis Reference	0004920
Problem Statement	HasWriterGroup and HasReaderGroup are missing.
Solution	Add: HasWriterGroup ReferenceType HasReaderGroup ReferenceType Change PubSubConnectionType: HasWriterGroup reference to WriterGroupName HasReaderGroup reference to ReaderGroupName

Topic	HasDictionaryEntry ReferenceType is not defined.
Errata Version	1.04.4
Spec Reference	TargetVersion="1.04.4" TargetPublicationDate="2019-11-20T00:00:00Z"
Mantis Reference	0004896
Problem Statement	HasDictionaryEntry ReferenceType is not defined.
Solution	Add: HasDictionaryEntry ReferenceType

Topic	Status Codes for FDI not defined.
Errata Version	1.04.4
Spec Reference	TargetVersion="1.04.4" TargetPublicationDate="2019-11-20T00:00:00Z"
Mantis Reference	0004800
Problem Statement	Status Codes for FDI not defined.
Solution	Add to the StatusCodes.csv: Good_Edited_DependentValueChanged Good_Edited_DominantValueChanged Good_Edited_DominantValueChanged_DependentValueChanged Bad_Edited_OutOfRange Bad_InitialValue_OutOfRange Bad_OutOfRange_DominantValueChanged Bad_Edited_OutOfRange_DominantValueChanged Bad_OutOfRange_DominantValueChanged_DependentValueChanged Bad_Edited_OutOfRange_DominantValueChanged_DependentValueChanged

Topic	ArrayDimensions should be specified properly for Array Arguments.
Errata Version	1.04.4
Spec Reference	TargetVersion="1.04.4" TargetPublicationDate="2019-11-20T00:00:00Z"
Mantis Reference	0004460
Problem Statement	ArrayDimensions should be specified properly for Array Arguments.

Solution	Change output for Array Arguments to:
	`<ArrayDimensions>`
	`  <UInt32>0</UInt32>`
	`</ArrayDimensions>`

Topic	TrustListOutOfDate and CertificateExpired Alarms missing on CertificateGroupType.
Errata Version	1.04.4
Spec Reference	TargetVersion="1.04.4" TargetPublicationDate="2019-11-20T00:00:00Z"
Mantis Reference	0005260
Problem Statement	TrustListOutOfDate and CertificateExpired Alarms missing on CertificateGroupType.
Solution	Add: TrustListOutOfDate AlarmType Add to CertificateGroupType: TrustListOutOfDate Component CertificateExpired Component

Topic	AuditHistoryAnnotationUpdateEventType Missing.
Errata Version	1.04.4
Spec Reference	TargetVersion="1.04.4" TargetPublicationDate="2019-11-20T00:00:00Z"
Mantis Reference	0005262
Problem Statement	AuditHistoryAnnotationUpdateEventType Missing.
Solution	Add: AuditHistoryAnnotationUpdateEventType

Topic	HistoryServerCapabilities ServerTimestampSupported Missing.
Errata Version	1.04.4
Spec Reference	TargetVersion="1.04.4" TargetPublicationDate="2019-11-20T00:00:00Z"
Mantis Reference	0005262
Problem Statement	HistoryServerCapabilities ServerTimestampSupported Missing
Solution	Add: HistoryServerCapabilities ServerTimestampSupported HistoricalDataConfigurationType ServerTimestampSupported HistoryServerCapabilitiesType ServerTimestampSupported

Topic	Add UpdateFrequency on TrustListType.
Errata Version	1.04.4
Spec Reference	TargetVersion="1.04.4" TargetPublicationDate="2019-11-20T00:00:00Z"
Mantis Reference	0005264
Problem Statement	Add UpdateFrequency on TrustListType
Solution	Add: Add UpdateFrequency on TrustListType

Topic	RoleSet ReferenceType Mismatch.
Errata Version	1.04.6
Spec Reference	TargetVersion="1.04.6" TargetPublicationDate="2020-04-14T00:00:00Z"
Mantis Reference	0005522
Problem Statement	Well known roles have wrong reference to RoleSet.

Solution	Update all Well Known Role Nodes:
	
	Change
	`<Reference ReferenceType="Organizes" IsForward="false">i=15606</Reference>`
	To
	`<Reference ReferenceType="HasComponent" IsForward="false">i=15606</Reference>`

Topic	RoleSet ReferenceType Mismatch.
Errata Version	1.04.6
Spec Reference	TargetVersion="1.04.6" TargetPublicationDate="2020-04-14T00:00:00Z"
Mantis Reference	0005522
Problem Statement	Well known roles have wrong reference to RoleSet.

Solution	Update all Well Known Role Nodes:
	
	Change
	`<Reference ReferenceType="Organizes" IsForward="false">i=15606</Reference>`
	To
	`<Reference ReferenceType="HasComponent" IsForward="false">i=15606</Reference>`

Topic	Property NodeVersion wrong ValueRank in BaseNodeSet2.xml.
Errata Version	1.04.6
Spec Reference	TargetVersion="1.04.6" TargetPublicationDate="2020-04-14T00:00:00Z"
Mantis Reference	0005525
Problem Statement	All well-known properties have wrong ValueRank when not an Array.

Solution	Update the following Property Nodes:
	• NodeVersion
	• ViewVersion
	• Icon
	• LocalTime
	• AllowNulls
	• ValueAsText
	• MaxStringLength
	• MaxCharacters
	• MaxByteStringLength
	• MaxArrayLength
	• EngineeringUnits
	
	Change
	`<UAVariable NodeId="i=11513" BrowseName="{name}" DataType="{datatype}" ValueRank="-2">`
	
	To
	`<UAVariable NodeId="i=11513" BrowseName="{name}" DataType="{datatype}">`

Topic	Nodeset wrong HasAddIn subtype of HasComponent.
Errata Version	1.04.6
Spec Reference	TargetVersion="1.04.6" TargetPublicationDate="2020-04-14T00:00:00Z"
Mantis Reference	0005501
Problem Statement	HasAddIn is a subtype of HasComponent and not a NonHierarchicalReferences.

Solution	Update HasAddIn:
	
	Change
	`<Reference ReferenceType="HasSubtype" IsForward="false">i=32</Reference>`
	
	To
	`<Reference ReferenceType="HasSubtype" IsForward="false">i=47</Reference>`

Topic	Binary schema does not mention attribute IsOptionSet.
Errata Version	1.04.6
Spec Reference	TargetVersion="1.04.6" TargetPublicationDate="2020-04-14T00:00:00Z"
Mantis Reference	0005413
Problem Statement	IsOptionSet is in the generated schema files but not in the schema definition
Solution	Update EnumeratedType in OPCBinarySchema.xsd: Add `<xs:attribute name="IsOptionSet" type="xs:boolean" default="false" />`

Topic	Abstract BaseTypes missing from NodeSet.
Errata Version	1.04.6
Spec Reference	TargetVersion="1.04.6" TargetPublicationDate="2020-04-14T00:00:00Z"
Mantis Reference	0005412
Problem Statement	There generation was omitting abstract basetypes with no fields.
Solution	Update UANodeSet.xsd: Now includes abstract base types with no fields. The definitions have an empty DataTypeDefinition element defined.

Topic	Add Types from Part 17.
Errata Version	1.04.6
Spec Reference	TargetVersion="1.04.6" TargetPublicationDate="2020-04-14T00:00:00Z"
Mantis Reference	0005261
Problem Statement	Types from Part 17 are not in NodeSet
Solution	Add all Part 17 types to NodeSet.

Topic	ValueRank mismatch between MultiStateDictionaryEntryDiscreteBaseType and its base type.
Errata Version	1.04.8
Spec Reference	TargetVersion="1.04.8" TargetPublicationDate="2020-11-20"
Mantis Reference	0006243
Problem Statement	MultiStateDictionaryEntryDiscreteBaseType ValueRank should be ‘Any’
Solution	Change MultiStateDictionaryEntryDiscreteBaseType ValueRank to ‘Any’

Topic	GetRejectedList method should take group id as an input argument
Errata Version	1.04.8
Spec Reference	TargetVersion="1.04.8" TargetPublicationDate="2020-11-20"
Mantis Reference	0006241
Problem Statement	Rejected Certificates lists are different for different CertificateGroups.
Solution	Added Optional GetRejectedList method to CertificateGroupType

Topic	Mismatch of reference types used for AliasNameCategoryType
Errata Version	1.04.8
Spec Reference	TargetVersion="1.04.8" TargetPublicationDate="2020-11-20"
Mantis Reference	0006036
Problem Statement	AliasNameCategoryType references to subcategories use HasComponent when it should be Organizes.
Solution	Change AliasNameCategoryType references to subcategories to Organizes.

Topic	Missing state and transition numbers.
Errata Version	1.04.8
Spec Reference	TargetVersion="1.04.8" TargetPublicationDate="2020-11-20"
Mantis Reference	0006099
Problem Statement	Missing state and transition numbers for FileTransferStateMachineType, ShelvedStateMachineType and ExclusiveLimitStateMachineType.
Solution	Add state and transition numbers for FileTransferStateMachineType, ShelvedStateMachineType and ExclusiveLimitStateMachineType.

Topic	DefaultInputValues is not used.
Errata Version	1.04.8
Spec Reference	TargetVersion="1.04.8" TargetPublicationDate="2020-11-20"
Mantis Reference	0006104
Problem Statement	DefaultInputValues is not used.
Solution	Removed DefaultInputValues.

Topic	New code Good_RetransmissionQueueNotSupported.
Errata Version	1.04.8
Spec Reference	TargetVersion="1.04.8" TargetPublicationDate="2020-11-20"
Mantis Reference	0006033
Problem Statement	New code Good_RetransmissionQueueNotSupported
Solution	Added code Good_RetransmissionQueueNotSupported

Topic	AccessRestrictionType DataType has wrong base type UInt32.
Errata Version	1.04.8
Spec Reference	TargetVersion="1.04.8" TargetPublicationDate="2020-11-20"
Mantis Reference	0006025
Problem Statement	AccessRestrictionType DataType has wrong base type UInt32
Solution	Changed AccessRestrictionType to UInt16.

Topic	AccessRestrictionType is missing flag ApplyRestrictionsToBrowse.
Errata Version	1.04.8
Spec Reference	TargetVersion="1.04.8" TargetPublicationDate="2020-11-20"
Mantis Reference	0006026
Problem Statement	AccessRestrictionType is missing flag ApplyRestrictionsToBrowse
Solution	Added ApplyRestrictionsToBrowse flag to AccessRestrictionType.

Topic	Add Amendment 4 - ECC Types.
Errata Version	1.04.8
Spec Reference	TargetVersion="1.04.8" TargetPublicationDate="2020-11-20"
Mantis Reference	0006259
Problem Statement	Types defined in Amendent 4 need to be added to NodeSet.
Solution	Added types defined in Amendent 4.

Topic	TypeNode and InstanceNode have the wrong Purpose in the UA nodeset file.
Errata Version	1.04.8
Spec Reference	TargetVersion="1.04.8" TargetPublicationDate="2020-11-20"
Mantis Reference	0006234
Problem Statement	TypeNode and InstanceNode have the wrong Purpose in the UA nodeset file
Solution	Purpose for InstanceNode and TypeNode is now CodeGenerator.

Topic	Wrong structure of ServerRedundancy node in Opc.Ua.NodeSet2.xml
Errata Version	1.04.8
Spec Reference	TargetVersion="1.04.8" TargetPublicationDate="2020-11-20"
Mantis Reference	0006231, 0006215
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. The Properties are still defined as unattached Nodes.

Topic	SessionServiceRequest.UrisVersion is Array in NodeSet when should be Scalar
Errata Version	1.04.8
Spec Reference	TargetVersion="1.04.8" TargetPublicationDate="2020-11-20"
Mantis Reference	0006268
Problem Statement	SessionServiceRequest.UrisVersion is Array in NodeSet when should be Scalar
Solution	SessionServiceRequest.UrisVersion changed to Scalar

Topic	CertificateExpired and TrustListOutOfDate MIssing in CertificateGroupType
Errata Version	1.04.9
Spec Reference	TargetVersion="1.04.9" TargetPublicationDate="2021-01-21"
Mantis Reference	0006397
Problem Statement	CertificateExpired and TrustListOutOfDate missing in Nodeset 1.04.8.
Solution	Add CertificateExpired and TrustListOutOfDate to CertificateGroupType.

Topic	Additional Status Codes needed for O-PAS
Errata Version	1.04.10
Spec Reference	TargetVersion="1.04.10" TargetPublicationDate="2021-07-15"
Mantis Reference	0005677
Problem Statement	O-PAS requires additional Uncertain and Good status codes to meet their requirements.
Solution	Add the following StatusCodes: UncertainTransducerInManual 0x42080000 UncertainSimulatedValue 0x42090000 UncertainSensorCalibration 0x420A0000 UncertainConfigurationError 0x420F0000 GoodCascadeInitializationAcknowledged 0x04010000 GoodCascadeInitializationRequest 0x04020000 GoodCascadeNotInvited 0x04030000 GoodCascadeNotSelected 0x04040000 GoodFaultStateActive 0x04070000 GoodInitiateFaultState 0x04080000 GoodCascade 0x04090000

Topic	Incorrect ReferenceType for AudibleSound Component.
Errata Version	1.04.10
Spec Reference	TargetVersion="1.04.10" TargetPublicationDate="2021-07-15"
Mantis Reference	0006413
Problem Statement	ReferenceType for AudibleSound Component of AlarmConditionType should be HasComponent.
Solution	Change Reference from AlarmConditionType to the AudibleSound Component to HasComponent.

Topic	DataType OptionSet is not abstract in Nodeset - Part 3 defines is as abstract
Errata Version	1.04.10
Spec Reference	TargetVersion="1.04.10" TargetPublicationDate="2021-07-15"
Mantis Reference	0006392
Problem Statement	DataType OptionSet is not abstract in Nodeset - Part 3 defines is as abstract
Solution	Change OptionSet.IsAbstract=TRUE.

Topic	IdentityCriteriaType Application value is missing.
Errata Version	1.04.10
Spec Reference	TargetVersion="1.04.10" TargetPublicationDate="2021-07-15"
Mantis Reference	0006392
Problem Statement	IdentityCriteriaType Application value is missing.
Solution	Add Application_7 to the IdentityCriteriaType enumeration.

Topic	ProgramDiagnostic2DataType is broken in Nodeset.
Errata Version	1.04.10
Spec Reference	TargetVersion="1.04.10" TargetPublicationDate="2021-07-15"
Mantis Reference	0006410
Problem Statement	ProgramDiagnostic2DataType is still using the broken StatusResult for LastMethodReturnStatus and not StatusCode like defined in the specification.
Solution	Change the ProgramDiagnostic2DataType.LastMethodReturnStatus DataType to StatusCode. Change the NodeId for the DataType Node and its associated Encoding from 15396...15405 to 24033...24042.

Topic	Problem Handling of StructureFields with abstract DataType
Errata Version	1.04.10
Spec Reference	TargetVersion="1.04.10" TargetPublicationDate="2021-07-15"
Mantis Reference	0006454
Problem Statement	It needs to be possible to distinguish between StructureFields that allow subtypes and those that do not allow subtypes.
Solution	Add StructureWithSubtypedValues and UnionWithSubtypedValues to the StructureType enumeration.

Topic	ValueRank mismatch between MultiStateDictionaryEntryDiscreteBaseType and its base type
Errata Version	1.04.10
Spec Reference	TargetVersion="1.04.10" TargetPublicationDate="2021-07-15"
Mantis Reference	0006244
Problem Statement	The ValueRank for MultistateDictionaryEntryDiscreteBaseType and MultistateDictionaryEntryDiscreteType needs to Any.
Solution	Change MultistateDictionaryEntryDiscreteBaseType ValueRank to “Any (-2)”. Change MultistateDictionaryEntryDiscreteType ValueRank to “Any (-2)”.

Topic	PubSubState does not contain PreOperational state
Errata Version	1.04.10
Spec Reference	TargetVersion="1.04.10" TargetPublicationDate="2021-07-15"
Mantis Reference	0006244
Problem Statement	PubSubState does not contain PreOperational state
Solution	Add PreOperational=4 to PubSubState Enumeration.

Topic	Wrong schema for SymbolicName simpleType.
Errata Version	1.04.10
Spec Reference	TargetVersion="1.04.10" TargetPublicationDate="2021-07-15"
Mantis Reference	0006108
Problem Statement	Schema for SymbolicName simpleType should not be an xs:list.

Solution	Replace schema for SymbolicName in UANodeSet.xsd with this:
	
	`<xs:simpleType name="SymbolicName">`
	`  <xs:restriction base="xs:string">`
	`  <xs:pattern value="[A-Za-z][A-Za-z0-9_]*" />`
	`  </xs:restriction>`
	`  </xs:simpleType>`

Topic	Deprecating DataTypeDictionary in the NodeSet.
Errata Version	1.04.10
Spec Reference	TargetVersion="1.04.10" TargetPublicationDate="2021-07-15"
Mantis Reference	0006536
Problem Statement	DataTypeDictionary and related types should be Deprecated.
Solution	Set the ReleaseStatus to Deprecated for: DataTypeDictionaryType DataTypeSystemType DataTypeDescriptionType and all instances of those types.

Topic	Add Part 22 content to NodeSet-File.
Errata Version	1.04.10
Spec Reference	TargetVersion="1.04.10" TargetPublicationDate="2021-07-15"
Mantis Reference	0006855
Problem Statement	Part 22 types need to be added to NodeSet.
Solution	Add add types from Part 22 (read Part 22 to get list).

Topic	Deprecate undocumented Date and Time DataTypes.
Errata Version	1.04.10
Spec Reference	TargetVersion="1.04.10" TargetPublicationDate="2021-07-15"
Mantis Reference	0007033
Problem Statement	Date and Time DataTypes are not defined in the specification and should not be in the NodeSet, however, they are used by some applications.
Solution	Mark Date and Time DataType as Deprecated.

Topic	SystemOffNormalAlarmType IsAbstract should be False.
Errata Version	1.04.10
Spec Reference	TargetVersion="1.04.10" TargetPublicationDate="2021-07-15"
Mantis Reference	0006998
Problem Statement	SystemOffNormalAlarmType IsAbstract should be False.
Solution	Change SystemOffNormalAlarmType IsAbstract to False.

Topic	Keep-alive message in JSON unclear or unfeasible.
Errata Version	1.04.10
Spec Reference	TargetVersion="1.04.10" TargetPublicationDate="2021-07-15"
Mantis Reference	0006996
Problem Statement	JSON DataSetMessage needs a way to distinguish between different message types.
Solution	Add MessageType bit to JsonDataSetMessageContentMask.

Topic	MaxByteStringLength property missing on FileType
Errata Version	1.04.10
Spec Reference	TargetVersion="1.04.10" TargetPublicationDate="2021-07-15"
Mantis Reference	0006812
Problem Statement	MaxByteStringLength property missing on FileType.
Solution	Add MaxByteStringLength Property to FileType.

Topic	Properties on DataTypes have ModelingRules.
Errata Version	1.04.10
Spec Reference	TargetVersion="1.04.10" TargetPublicationDate="2021-07-15"
Mantis Reference	0006614
Problem Statement	Properties on DataTypes have ModelingRules that should not be there.
Solution	Changed the ModelCompiler to remove ModellingRules on all Properties of DataTypes, RefrenceTypes and Views.

Topic	Wrong BrowserName for Server_ServerRedundancy components.
Errata Version	1.04.10
Spec Reference	TargetVersion="1.04.10" TargetPublicationDate="2021-07-15"
Mantis Reference	0006550
Problem Statement	Wrong BrowserName for Server_ServerRedundancy components.
Solution	Fix BrowseName for: Server_ServerRedundancy_CurrentServerId Server_ServerRedundancy_RedundantServerArray Server_ServerRedundancy_ServerUrl Server_ServerRedundancy_ServerNetworkGroupsiArray

Topic	Wrong text for TestingConditionClassType.
Errata Version	1.04.11
Spec Reference	TargetVersion="1.04.11" TargetPublicationDate="2022-03-29"
Mantis Reference	0007429
Problem Statement	TestingConditionSubClassType is the wrong name.
Solution	Change: TestingConditionSubClassType to TestingConditionClassType

Topic	Draft Identifiers and DataTypeDescriptions appear in NodeSet.
Errata Version	1.04.11
Spec Reference	TargetVersion="1.04.11" TargetPublicationDate="2021-TBD"
Mantis Reference	0007440
Problem Statement	Numerous of Identifiers and DataTypeDescriptions from unreleased Nodes appear in the NodeSet that should not be there.
Solution	Update files: Opc.Ua.NodeSet2.Services.xml Opc.Ua.NodeSet2.xml NodeIds.csv With the files from here: https://files.opcfoundation.org/schemas/UA/1.04/

Topic	No way to specify XML schema namespace in NodeSet.
Errata Version	1.04.11
Spec Reference	TargetVersion="1.04.11" TargetPublicationDate="2022-03-29"
Mantis Reference	0007441
Problem Statement	No way to specify XML schema namespace in NodeSet.
Solution	Add XmlSchemaUri ModelTableEntry element in UANodeSet.xsd Updated files are here: https://files.opcfoundation.org/schemas/UA/1.04/

Topic	Node CertificateErrorEventId missing in NodeSet 1.04
Errata Version	1.04.11
Spec Reference	TargetVersion="1.04.11" TargetPublicationDate="2022-03-29"
Mantis Reference	0007484
Problem Statement	Node CertificateErrorEventId missing in NodeSet 1.04.
Solution	Add CertificateErrorEventId to AuditOpenSecureChannelEventType.

Topic	Clarification on NamingRule.
Errata Version	1.04.11
Spec Reference	TargetVersion="1.04.11" TargetPublicationDate="2022-03-29"
Mantis Reference	0007904
Problem Statement	NamingRule is obsolete and no longer serves any purpose.
Solution	Remove.NamingRule Property from all ModellingRules.

Topic	HistoryEventDetails should not have a NodeId.
Errata Version	1.04.12
Spec Reference	Part 6 Section 5.1 – DataEncoding General
Mantis Reference	0009063
Problem Statement	HistoryEventDetails should be an abstract supertype with no members.
Solution	Move NodeId field into subtypes.

_____