

OPC Unified Architecture

Specification

Part 7: Profiles

Release 1.03

December 11, 2015

| | | | | |
|---------------------|--------------------------|-----------|--|--|
| Specification Type: | Industry Specification | Standard | Comments: | |
| Title: | OPC Unified Architecture | Date: | December 11, 2015 | |
| | Part 7 - Profiles | | | |
| Version: | Release 1.03 | Software: | MS-Word | |
| | | Source: | OPC UA Part 7 - Profiles 1.03 Specification.docx | |
| Author: | OPC Foundation | Status: | Release | |
| | | | | |

CONTENTS

Page

| | |
|---|------|
| FIGURES | vii |
| TABLES | viii |
| FOREWORD | xii |
| AGREEMENT OF USE | xii |
| 1 Scope | 1 |
| 2 Normative references | 1 |
| 3 Terms, definitions, and conventions | 2 |
| 3.1 Terms and definitions | 2 |
| 3.2 Abbreviations | 3 |
| 4 Overview | 3 |
| 4.1 General | 3 |
| 4.2 ConformanceUnit | 4 |
| 4.3 Profiles | 4 |
| 4.4 Profile Categories | 5 |
| 5 ConformanceUnits | 5 |
| 5.1 Overview | 5 |
| 5.2 Services | 6 |
| 5.3 Transport and communication related features | 15 |
| 5.4 Information Model and AddressSpace related features | 21 |
| 5.5 Miscellaneous | 39 |
| 6 Profiles | 40 |
| 6.1 Overview | 40 |
| 6.2 Profile list | 40 |
| 6.3 Conventions for Profile definitions | 45 |
| 6.4 Applications | 45 |
| 6.5 Profile tables | 47 |
| 6.5.1 Introduction | 47 |
| 6.5.2 Core Server Facet | 47 |
| 6.5.3 Global Certificate Management Server Facet | 48 |
| 6.5.4 Subnet Discovery Server Facet | 48 |
| 6.5.5 Base Server Behaviour Facet | 48 |
| 6.5.6 Request State Change Server Facet | 48 |
| 6.5.7 Attribute WriteMask Server Facet | 48 |
| 6.5.8 File Access Server Facet | 49 |
| 6.5.9 Documentation Server Facet | 49 |
| 6.5.10 Embedded DataChange Subscription Server Facet | 49 |
| 6.5.11 Standard DataChange Subscription Server Facet | 49 |
| 6.5.12 Enhanced DataChange Subscription Server Facet | 50 |
| 6.5.13 Durable Subscription Server Facet | 51 |
| 6.5.14 Data Access Server Facet | 51 |
| 6.5.15 ComplexType Server Facet | 51 |
| 6.5.16 Standard Event Subscription Server Facet | 52 |
| 6.5.17 Address Space Notifier Server Facet | 52 |
| 6.5.18 A & C Base Condition Server Facet | 52 |

| | | |
|--------|--|----|
| 6.5.19 | A & C Refresh2 Server Facet | 53 |
| 6.5.20 | A & C Address Space Instance Server Facet..... | 53 |
| 6.5.21 | A & C Enable Server Facet | 53 |
| 6.5.22 | A & C Alarm Server Facet | 53 |
| 6.5.23 | A & C Acknowledgeable Alarm Server Facet | 54 |
| 6.5.24 | A & C Exclusive Alarming Server Facet..... | 54 |
| 6.5.25 | A & C Non-Exclusive Alarming Server Facet | 54 |
| 6.5.26 | A & C Previous Instances Server Facet..... | 54 |
| 6.5.27 | A & C Dialog Server Facet | 55 |
| 6.5.28 | A & C CertificateExpiration Server Facet..... | 55 |
| 6.5.29 | A & E Wrapper Facet | 55 |
| 6.5.30 | Method Server Facet..... | 56 |
| 6.5.31 | Auditing Server Facet..... | 57 |
| 6.5.32 | Node Management Server Facet | 57 |
| 6.5.33 | Client Redundancy Server Facet | 57 |
| 6.5.34 | Redundancy Transparent Server Facet | 57 |
| 6.5.35 | Redundancy Visible Server Facet..... | 58 |
| 6.5.36 | Historical Raw Data Server Facet..... | 58 |
| 6.5.37 | Historical Aggregate Server Facet | 58 |
| 6.5.38 | Historical Access Structured Data Server Facet | 59 |
| 6.5.39 | Historical Data AtTime Server Facet | 59 |
| 6.5.40 | Historical Access Modified Data Server Facet | 60 |
| 6.5.41 | Historical Annotation Server Facet | 60 |
| 6.5.42 | Historical Data Update Server Facet | 60 |
| 6.5.43 | Historical Data Replace Server Facet..... | 60 |
| 6.5.44 | Historical Data Insert Server Facet..... | 61 |
| 6.5.45 | Historical Data Delete Server Facet | 61 |
| 6.5.46 | Base Historical Event Server Facet | 61 |
| 6.5.47 | Historical Event Update Server Facet..... | 61 |
| 6.5.48 | Historical Event Replace Server Facet | 61 |
| 6.5.49 | Historical Event Insert Server Facet | 62 |
| 6.5.50 | Historical Event Delete Server Facet..... | 62 |
| 6.5.51 | Aggregate Subscription Server Facet | 62 |
| 6.5.52 | Nano Embedded Device Server Profile..... | 63 |
| 6.5.53 | Micro Embedded Device Server Profile | 63 |
| 6.5.54 | Embedded UA Server Profile..... | 64 |
| 6.5.55 | Standard UA Server Profile | 64 |
| 6.5.56 | Global Discovery Server Profile..... | 64 |
| 6.5.57 | Global Discovery and Certificate Management Server Profile | 65 |
| 6.5.58 | Core Client Facet | 65 |
| 6.5.59 | Request State Change Client Facet | 65 |
| 6.5.60 | Global Certificate Management Client Facet..... | 66 |
| 6.5.61 | Base Client Behaviour Facet | 66 |
| 6.5.62 | Discovery Client Facet | 66 |
| 6.5.63 | Subnet Discovery Client Facet | 66 |
| 6.5.64 | Global Discovery Client Facet | 67 |
| 6.5.65 | AddressSpace Lookup Client Facet..... | 67 |
| 6.5.66 | Entry-Level Support Client Facet..... | 67 |
| 6.5.67 | Entry Level Support 2015 Client Facet | 67 |

| | | |
|---------|--|----|
| 6.5.68 | Multi-Server Client Connection Facet | 68 |
| 6.5.69 | File Access Client Facet..... | 68 |
| 6.5.70 | Documentation – Client | 68 |
| 6.5.71 | Attribute Read Client Facet | 68 |
| 6.5.72 | Attribute Write Client Facet | 68 |
| 6.5.73 | DataChange Subscriber Client Facet..... | 69 |
| 6.5.74 | Durable Subscription Client Facet | 69 |
| 6.5.75 | DataAccess Client Facet | 69 |
| 6.5.76 | Event Subscriber Client Facet..... | 69 |
| 6.5.77 | Base Event Processing Client Facet..... | 70 |
| 6.5.78 | Notifier and Source Hierarchy Client Facet..... | 70 |
| 6.5.79 | A & C Base ConditionClient Facet | 70 |
| 6.5.80 | A & C Refresh2 Client Facet | 70 |
| 6.5.81 | A & C Address Space Instance Client Facet..... | 71 |
| 6.5.82 | A & C Enable Client Facet..... | 71 |
| 6.5.83 | A & C Alarm Client Facet | 71 |
| 6.5.84 | A & C Exclusive Alarming Client Facet | 71 |
| 6.5.85 | A & C Non-Exclusive Alarming Client Facet..... | 72 |
| 6.5.86 | A & C Previous Instances Client Facet | 72 |
| 6.5.87 | A & C Dialog Client Facet..... | 72 |
| 6.5.88 | A & C CertificateExpiration Client Facet | 72 |
| 6.5.89 | A & E Proxy Facet..... | 72 |
| 6.5.90 | Method Client Facet | 73 |
| 6.5.91 | Auditing Client Facet | 74 |
| 6.5.92 | Node Management Client Facet | 74 |
| 6.5.93 | Advanced Type Programming Client Facet..... | 74 |
| 6.5.94 | Diagnostic Client Facet | 74 |
| 6.5.95 | Redundant Client Facet..... | 74 |
| 6.5.96 | Redundancy Switch Client Facet | 75 |
| 6.5.97 | Historical Access Client Facet..... | 75 |
| 6.5.98 | Historical Annotation Client Facet | 75 |
| 6.5.99 | Historical Data AtTime Client Facet..... | 75 |
| 6.5.100 | Historical Aggregate Client Facet | 75 |
| 6.5.101 | Historical Data Update Client Facet..... | 76 |
| 6.5.102 | Historical Data Replace Client Facet | 76 |
| 6.5.103 | Historical Data Insert Client Facet | 77 |
| 6.5.104 | Historical Data Delete Client Facet..... | 78 |
| 6.5.105 | Historical Access Client Server Timestamp Facet..... | 78 |
| 6.5.106 | Historical Access Modified Data Client Facet | 78 |
| 6.5.107 | Structured Data AtTime Client Facet | 78 |
| 6.5.108 | Historical Structured Data Access Client Facet | 78 |
| 6.5.109 | Historical Structured Data Modified Client Facet | 79 |
| 6.5.110 | Historical Structured Data Delete Client Facet..... | 79 |
| 6.5.111 | Historical Structured Data Update Client Facet | 79 |
| 6.5.112 | Historical Structured Data Replace Client Facet | 79 |
| 6.5.113 | Historical Structured Data Insert Client Facet | 79 |
| 6.5.114 | Historical Events Client Facet | 79 |
| 6.5.115 | Historical Event Update Client Facet | 80 |
| 6.5.116 | Historical Event Replace Client Facet | 80 |

| | | |
|---------|--|----|
| 6.5.117 | Historical Event Delete Client Facet | 80 |
| 6.5.118 | Historical Event Insert Client Facet | 80 |
| 6.5.119 | Aggregate Subscriber Client Facet | 81 |
| 6.5.120 | Global Certificate Management Client Profile | 82 |
| 6.5.121 | Standard UA Client Profile | 82 |
| 6.5.122 | User Token – Anonymous Facet..... | 83 |
| 6.5.123 | User Token – User Name Password Server Facet | 83 |
| 6.5.124 | User Token – X509 Certificate Server Facet..... | 83 |
| 6.5.125 | User Token – Issued Token Server Facet..... | 83 |
| 6.5.126 | User Token – Issued Token Windows Server Facet..... | 83 |
| 6.5.127 | User Token – User Name Password Client Facet | 83 |
| 6.5.128 | User Token – X509 Certificate Client Facet..... | 84 |
| 6.5.129 | User Token – Issued Token Client Facet | 84 |
| 6.5.130 | User Token – Issued Token Windows Client Facet | 84 |
| 6.5.131 | UA-TCP UA-SC UA Binary | 84 |
| 6.5.132 | SOAP-HTTP WS-SC UA XML..... | 84 |
| 6.5.133 | SOAP-HTTP WS-SC UA Binary..... | 84 |
| 6.5.134 | SOAP-HTTP WS-SC UA XML-UA Binary..... | 84 |
| 6.5.135 | HTTPS UA Binary | 85 |
| 6.5.136 | HTTPS UA XML | 85 |
| 6.5.137 | Security User Access Control Full | 85 |
| 6.5.138 | Security User Access Control Base | 85 |
| 6.5.139 | Security Time Synchronization | 85 |
| 6.5.140 | Best Practice – Audit Events | 86 |
| 6.5.141 | Best Practice – Alarm Handling | 86 |
| 6.5.142 | Best Practice – Random Numbers | 86 |
| 6.5.143 | Best Practice – Timeouts | 86 |
| 6.5.144 | Best Practice – Administrative Access..... | 86 |
| 6.5.145 | Best Practice – Strict Message Handling | 87 |
| 6.5.146 | Best Practice – Audit Events Client | 87 |
| 6.5.147 | SecurityPolicy – None | 87 |
| 6.5.148 | SecurityPolicy – Basic128Rsa15 | 87 |
| 6.5.149 | SecurityPolicy – Basic256 | 88 |
| 6.5.150 | SecurityPolicy – Basic256Sha256 | 88 |
| 6.5.151 | TransportSecurity – TLS 1.0..... | 88 |
| 6.5.152 | TransportSecurity – TLS 1.1..... | 88 |
| 6.5.153 | TransportSecurity – TLS 1.2..... | 88 |
| 6.5.154 | TransportSecurity – TLS 1.2 with PFS..... | 89 |
| | Bibliography | 90 |

FIGURES

Figure 1 – Profile – ConformanceUnit – TestCases 4

Figure 2 – HMI Client sample 45

Figure 3 – Embedded Server sample..... 46

Figure 4 – Standard UA Server sample 46

TABLES

| | |
|--|----|
| Table 1 – ProfileCategories | 5 |
| Table 2 – ConformanceGroups | 5 |
| Table 3 – Discovery Services | 6 |
| Table 4 – Session Services | 8 |
| Table 5 – Node Management Services | 9 |
| Table 6 – View Services | 10 |
| Table 7 – Attribute Services | 10 |
| Table 8 – Method Services | 12 |
| Table 9 – Monitored Item Services | 12 |
| Table 10 – Subscription Services | 14 |
| Table 11 – Security | 15 |
| Table 12 – Protocol and Encoding | 20 |
| Table 13 – Base information | 21 |
| Table 14 – Address Space model | 23 |
| Table 15 – Data Access | 24 |
| Table 16 – Alarms and Conditions | 25 |
| Table 17 – Historical Access | 27 |
| Table 18 – Aggregates | 30 |
| Table 19 – Auditing | 38 |
| Table 20 – Redundancy | 38 |
| Table 21 – Global Discovery Server | 39 |
| Table 22 – Miscellaneous | 39 |
| Table 23 – Profile list | 40 |
| Table 24 – Core Server Facet | 47 |
| Table 25 – Global Certificate Management Server Facet | 48 |
| Table 26 – Subnet Discovery Server Facet | 48 |
| Table 27 – Base Server Behaviour Facet | 48 |
| Table 28 – Request State Change Server Facet | 48 |
| Table 29 – Attribute WriteMask Server Facet | 49 |
| Table 30 – File Access Server Facet | 49 |
| Table 31 – Documentation Server Facet | 49 |
| Table 32 – Embedded DataChange Subscription Server Facet | 49 |
| Table 33 – Standard DataChange Subscription Server Facet | 50 |
| Table 34 – Enhanced DataChange Subscription Server Facet | 51 |
| Table 35 – Durable Subscription Server Facet | 51 |
| Table 36 – Data Access Server Facet | 51 |
| Table 37 – ComplexType Server Facet | 51 |
| Table 38 – Standard Event Subscription Server Facet | 52 |
| Table 39 – Address Space Notifier Server Facet | 52 |
| Table 40 – A & C Base Condition Server Facet | 53 |
| Table 41 – A & C Refresh2 Server Facet | 53 |
| Table 42 – A & C Address Space Instance Server Facet | 53 |

| | |
|---|----|
| Table 43 – A & C Enable Server Facet | 53 |
| Table 44 – A & C Alarm Server Facet | 53 |
| Table 45 – A & C Acknowledgeable Alarm Server Facet | 54 |
| Table 46 – A & C Exclusive Alarming Server Facet | 54 |
| Table 47 – A & C Non-Exclusive Alarming Server Facet | 54 |
| Table 48 – A & C Previous Instances Server Facet | 55 |
| Table 49 – A & C Dialog Server Facet | 55 |
| Table 50 – A & C CertificateExpiration Server Facet | 55 |
| Table 51 – A & E Wrapper Facet | 56 |
| Table 52 – Method Server Facet | 57 |
| Table 53 – Auditing Server Facet | 57 |
| Table 54 – Node Management Server Facet | 57 |
| Table 55 – Client Redundancy Server Facet | 57 |
| Table 56 – Redundancy Transparent Server Facet | 58 |
| Table 57 – Redundancy Visible Server Facet | 58 |
| Table 58 – Historical Raw Data Server Facet | 58 |
| Table 59 – Historical Aggregate Server Facet | 58 |
| Table 60 – Historical Access Structured Data Server Facet | 59 |
| Table 61 – Historical Data AtTime Server Facet | 59 |
| Table 62 – Historical Access Modified Data Server Facet | 60 |
| Table 63 – Historical Annotation Server Facet | 60 |
| Table 64 – Historical Data Update Server Facet | 60 |
| Table 65 – Historical Data Replace Server Facet | 61 |
| Table 66 – Historical Data Insert Server Facet | 61 |
| Table 67 – Historical Data Delete Server Facet | 61 |
| Table 68 – Base Historical Event Server Facet | 61 |
| Table 69 – Historical Event Update Server Facet | 61 |
| Table 70 – Historical Event Replace Server Facet | 62 |
| Table 71 – Historical Event Insert Server Facet | 62 |
| Table 72 – Historical Event Delete Server Facet | 62 |
| Table 73 – Aggregate Subscription Server Facet | 62 |
| Table 74 – Nano Embedded Device Server Profile | 63 |
| Table 75 – Micro Embedded Device Server Profile | 63 |
| Table 76 – Embedded UA Server Profile | 64 |
| Table 77 – Standard UA Server Profile | 64 |
| Table 78 – Global Discovery Server Profile | 64 |
| Table 79 – Global Discovery and Certificate Management Server Profile | 65 |
| Table 80 – Core Client Facet | 65 |
| Table 81 – Request State Change Client Facet | 65 |
| Table 82 – Global Certificate Management Client Facet | 66 |
| Table 83 – Base Client Behaviour Facet | 66 |
| Table 84 – Discovery Client Facet | 66 |
| Table 85 – Subnet Discovery Client Facet | 67 |

| | |
|---|----|
| Table 86 – Global Discovery Client Facet | 67 |
| Table 87 – AddressSpace Lookup Client Facet | 67 |
| Table 88 – Entry Level Support 2015 Client Facet | 67 |
| Table 89 – Multi-Server Client Connection Facet | 68 |
| Table 90 – File Access Client Facet | 68 |
| Table 91 – Documentation – Client | 68 |
| Table 92 – Attribute Read Client Facet | 68 |
| Table 93 – Attribute Write Client Facet | 68 |
| Table 94 – DataChange Subscriber Client Facet | 69 |
| Table 95 – Durable Subscription Client Facet | 69 |
| Table 96 – DataAccess Client Facet | 69 |
| Table 97 – Event Subscriber Client Facet | 69 |
| Table 98 – Base Event Processing Client Facet | 70 |
| Table 99 – Notifier and Source Hierarchy Client Facet | 70 |
| Table 100 – A & C Base Condition Client Facet | 70 |
| Table 101 – A & C Reresh2 Client Facet | 71 |
| Table 102 – A & C Address Space Instance Client Facet | 71 |
| Table 103 – A & C Enable Client Facet | 71 |
| Table 104 – A & C Alarm Client Facet | 71 |
| Table 105 – A & C Exclusive Alarming Client Facet | 71 |
| Table 106 – A & C Non-Exclusive Alarming Client Facet | 72 |
| Table 107 – A & C Previous Instances Client Facet | 72 |
| Table 108 – A & C Dialog Client Facet | 72 |
| Table 109 – A & C CertificateExpiration Client Facet | 72 |
| Table 110 – A & E Proxy Facet | 73 |
| Table 111 – Method Client Facet | 73 |
| Table 112 – Auditing Client Facet | 74 |
| Table 113 – Node Management Client Facet | 74 |
| Table 114 – Advanced Type Programming Client Facet | 74 |
| Table 115 – Diagnostic Client Facet | 74 |
| Table 116 – Redundant Client Facet | 74 |
| Table 117 – Redundancy Switch Client Facet | 75 |
| Table 118 – Historical Access Client Facet | 75 |
| Table 119 – Historical Annotation Client Facet | 75 |
| Table 120 – Historical Data AtTime Client Facet | 75 |
| Table 121 – Historical Aggregate Client Facet | 75 |
| Table 122 – Historical Data Update Client Facet | 76 |
| Table 123 – Historical Data Replace Client Facet | 76 |
| Table 124 – Historical Data Insert Client Facet | 78 |
| Table 125 – Historical Data Delete Client Facet | 78 |
| Table 126 – Historical Access Client Server Timestamp Facet | 78 |
| Table 127 – Historical Access Modified Data Client Facet | 78 |
| Table 128 – Historical Structured Data AtTime Client Facet | 78 |

| | |
|--|----|
| Table 129 – Historical Structured Data Access Client Facet | 78 |
| Table 130 – Historical Structured Data Modified Client Facet | 79 |
| Table 131 – Historical Structured Data Delete Client Facet | 79 |
| Table 132 – Historical Structured Data Update Client Facet | 79 |
| Table 133 – Historical Structured Data Replace Client Facet..... | 79 |
| Table 134 – Historical Structured Data Insert Client Facet..... | 79 |
| Table 135 – Historical Events Client Facet | 80 |
| Table 136 – Historical Event Update Client Facet..... | 80 |
| Table 137 – Historical Event Replace Client Facet | 80 |
| Table 138 – Historical Event Delete Client Facet..... | 80 |
| Table 139 – Historical Event Insert Client Facet | 81 |
| Table 140 – Aggregate Subscriber Client Facet..... | 81 |
| Table 141 – Global Certificate Management Client Profile | 82 |
| Table 142 – Standard UA Client Profile | 82 |
| Table 143 – User Token – Anonymous Facet | 83 |
| Table 144 – User Token – User Name Password Server Facet..... | 83 |
| Table 145 – User Token – X509 Certificate Server Facet | 83 |
| Table 146 – User Token – Issued Token Server Facet | 83 |
| Table 147 – User Token – Issued Token Windows Server Facet..... | 83 |
| Table 148 – User Token – User Name Password Client Facet | 84 |
| Table 149 – User Token – X509 Certificate Client Facet..... | 84 |
| Table 150 – User Token – Issued Token Client Facet..... | 84 |
| Table 151 – User Token – Issued Token Windows Client Facet..... | 84 |
| Table 152 – UA-TCP UA-SC UA Binary | 84 |
| Table 153 – HTTPS UA Binary | 85 |
| Table 154 – HTTPS UA XML | 85 |
| Table 155 – Security User Access Control Full | 85 |
| Table 156 – Security User Access Control Base..... | 85 |
| Table 157 – Security Time Synchronization | 86 |
| Table 158 – Best Practice – Audit Events | 86 |
| Table 159 – Best Practice – Alarm Handling..... | 86 |
| Table 160 – Best Practice – Random Numbers..... | 86 |
| Table 161 – Best Practice – Timeouts | 86 |
| Table 162 – Best Practice – Administrative Access | 86 |
| Table 163 – Best Practice – Strict Message Handling..... | 87 |
| Table 164 – Best Practice – Audit Events Client..... | 87 |
| Table 165 – SecurityPolicy – None..... | 87 |
| Table 166 – SecurityPolicy – Basic128Rsa15 | 87 |
| Table 167 – SecurityPolicy – Basic256..... | 88 |
| Table 168 – SecurityPolicy – Basic256Sha256 | 88 |
| Table 169 – TransportSecurity – TLS 1.2 | 89 |
| Table 170 – TransportSecurity – TLS 1.2 with PFS | 89 |

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 © 2006-2015, 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 FOUNDATION 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 of 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>.

Revision 1.03 Highlights

The following table includes the Mantis issues resolved with this revision.

| Mantis ID | Summary | Resolution |
|-----------|--|--|
| 2652 | Missing option "Accept any valid instance certificate" in Base Behaviour facet (both Client and Server) | Added option "Accept any valid instance certificate" to Security Administration CU. This CU is included in the Base Behaviour facets. |
| 2643 | A security level is defined in Part 4 to specify the relative level of endpoints to each other. This security level may change when security profiles get deprecated. A fixed level defined within a profile therefore makes no sense. | The security level conformance units have been removed. |
| 2474 | The description for the CU " Historical Access Data Max Nodes Read Continuation Point " falsely states that the MaxNodesPerHistoryRead property is under the ServerCapabilities object | Changed text to state that this property is actually under the ServerCapabilities.OperationLimits. |
| 2468 | We currently have a CU defined for insert and another for update, but not for replace. | Changed the description of the "Update Value" CU into "supports updating ...". Added a "Replace Value" CU to the same facet (optional). Also added a replace CU to the corresponding Client facet. |
| 2378 | FullFeatured Profile is used like a term but not defined | Added a proper term. |
| 2829 | Add Conformance Units for Subnet Discovery | Added Register2, FindServersOnNetwork and mDNS Publishing (for Servers with no LDS-ME). Added CUs as optional to appropriate Profiles. |
| 2640 | Add GDS Profiles | Added CUs and a new facet for UA Servers that implement the GDS Information Model. Added CU for Clients to find Servers using a GDS. Added CUs for a Global Discovery Server and a Global Certificate Manager. Created full featured Profiles for a GDS and a global Certificate Manager. |
| 2673 | Nano embedded server requires encryption | Specified rules for encryption in user token conformance units. |
| 2900 | Durable subscriptions | Added CUs and Facets for durable subscriptions. |
| 2911 | Deprecate WS Secure Conversation | Kept the headers but removed the description and added "Note: Deprecated in Version 1.03 because WS-SecureConversation has not been widely adopted by industry". |
| 3011 | New Alarm for certificate expiration | Added CUs and Facets for this new alarm type. |
| 3009 | Refresh2 (refresh individual monitored items) | Added CUs and Facets for Refresh2. |
| 2778 | "Monitor Complex Event Filter" is exclusively for "TypeOf"? | Clarified issue by changing the description into Support for the 'TypeOf' complex Event filter operator. |
| 3048 | Add A&C "SystemOffNormal" Conformance Units | Created CUs for SystemOffNormal and added to proper profiles. |
| 2654 | A&C Profile descriptions are lacking clarity | Improved descriptions based on proposals from compliance group. |
| 3044 | Entry-level support Client CU needs additional text. | Replaced Entry Level support Client facet by new version. |
| 3043 | Client profiles need at least one full-featured profile. | Created Standard UA Client Profile. |

| Mantis ID | Summary | Resolution |
|-----------|--|---|
| 2357 | Need CUs and Profiles for remote Nodes. | Created CUs for Browse and Attribute Access. Added CUs to proper Profiles. |
| 3070 | Events should not be in AddressSpaceLookup Facet. | Created Base Event Processing Client Facet. Removed Events from AddressSpaceLookup Client Facet. |
| 2777 | Underlying system – clarifications needed | Created new CUs for SystemStatusChangeEvents and DeviceFailureEvents. Added these CUs to proper Facets. |
| 3065 | Need better rules for subscriptions in Micro Devices | Added text to the CUs that “the size of the MonitoredItem is less than or equal to size of Double”. |
| 3088 | Need CU and Facet for RequestServerStateChange Method | Created CUs and Facets for Server and Client. |
| 3122 | All Security Policy Profiles must require minimum hash for all issuers in the chain. | Added text to the CUs stating the minimum hash required for signing of any certificate in the chain. |
| 3121 | Deprecate TLS 1.0 and 1.1 since RC4 is not considered secure anymore. | Deprecated both facets |
| 3123 | Add TLS 1.2 Profile with PFS (Perfect Forward Secrecy). | Added a new facet as requested. |
| | | |

OPC UNIFIED ARCHITECTURE –

Part 7: Profiles

1 Scope

This part describes the OPC Unified Architecture (OPC UA) *Profiles*. The *Profiles* in this document are used to segregate features with regard to testing of OPC UA products and the nature of the testing (tool based or lab based). This includes the testing performed by the OPC Foundation provided OPC UA CTT (a self-test tool) and by the OPC Foundation provided Independent certification test labs. This could equally as well refer to test tools provided by another organization or a test lab provided by another organization. What is important is the concept of automated tool based testing versus lab based testing. The scope of this standard includes defining functionality that can only be tested in an a lab and defining the grouping of functionality that is to be used when testing OPC UA products either in a lab or using automated tools. The definition of actual *TestCases* is not within the scope of this document, but the general categories of *TestCases* are within the scope of this document.

Most OPC UA applications will conform to several, but not all of the *Profiles*.

2 Normative references

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

Part 1: OPC UA Specification: Part 1 – Concepts.

<http://www.opcfoundation.org/UA/Part1/>

Part 2: OPC UA Specification: Part 2 – Security Model

<http://www.opcfoundation.org/UA/Part2/>

Part 3: OPC UA Specification: Part 3 – Address Space Model

<http://www.opcfoundation.org/UA/Part3/>

Part 4: OPC UA Specification: Part 4 – Service

<http://www.opcfoundation.org/UA/Part4/>

Part 5: OPC UA Specification: Part 5 – Information Model

<http://www.opcfoundation.org/UA/Part5/>

Part 6: OPC UA Specification: Part 6 – Mapping

<http://www.opcfoundation.org/UA/Part6/>

Part 8: OPC UA Specification: Part 8 – Data Access

<http://www.opcfoundation.org/UA/Part8/>

Part 9: OPC UA Specification: Part 9 – Alarms and Conditions

<http://www.opcfoundation.org/UA/Part9/>

Part 10: OPC UA Specification: Part 10 – Programs

<http://www.opcfoundation.org/UA/Part10/>

Part 11: OPC UA Specification: Part 11 – Historical Access

<http://www.opcfoundation.org/UA/Part11/>

Part 12: OPC UA Specification: Part 12 – Discovery

<http://www.opcfoundation.org/UA/Part12/>

Part 13: OPC UA Specification: Part 13 – Aggregates

<http://www.opcfoundation.org/UA/Part13/>

3 Terms, definitions, and conventions

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in Part 1, Part 2, Part 3, Part 4, Part 6, and Part 8 as well as the following apply. An overview of the terms defined in this standard and their interaction can be viewed in Figure 1.

3.1.1

application

a software program that executes or implements some aspect of OPC UA

Note 1 to entry: The application could run on any machine and perform any function. The application could be software or it could be a hardware application, the only requirement is that it implements OPC UA.

3.1.2

ConformanceUnit

a specific set of OPC UA features that can be tested as a single entity

Note 1 to entry: A *ConformanceUnit* can cover a group of services, portions of services or information models. For additional detail see Clause 5.

3.1.3

ConformanceGroup

a group of *ConformanceUnits* that is given a name

Note 1 to entry: This grouping is only to assist in organizing *ConformanceUnits*. Typical *ConformanceGroups* include groups for each of the service sets in OPC UA and each of the Information Model standards.

3.1.4

Facet

a *Profile* dedicated to a specific feature that a *Server* or *Client* may require

Note 1 to entry: *Facets* are typically combined to form higher-level *Profiles*. The use of the term *Facet* in the title of a *Profile* indicates that the given *Profile* is not a standalone *Profile*.

3.1.5

FullFeatured Profile

a *Profile* that defines all features necessary to build a functional OPC UA *Application*

Note 1 to entry: A *FullFeatured Profile* in particular adds definitions of the transport and security requirements.

3.1.6

ProfileCategory

arranges *Profiles* into application classes, such as *Server* or *Client*

Note 1 to entry: These categories help determine the type of *Application* that a given *Profile* would be used for. For additional details see 4.4.

3.1.7

TestCase

a technical description of a set of steps required to test a particular function or information model

Note 1 to entry: *TestCases* provide sufficient details to allow a developer to implement them in code. *TestCases* also provide a detailed summary of the expected result(s) from the execution of the implemented code and any precondition(s) that must be established before the *TestCase* can be executed.

3.1.8

TestLab

a facility that is designated to provide testing services

Note 1 to entry: These services include but are not limited to personal that directly perform testing, automated testing and a formal repeatable process. The OPC Foundation has provided detailed standard describing OPC UA TestLabs and the testing they are to provided (see Compliance Part 8 UA *Server*, Compliance Part 9 UA *Client*).

3.2 Abbreviations

| | |
|------|---|
| DA | Data Access |
| HA | Historical Access |
| HMI | Human Machine Interface |
| NIST | National Institute of Standard and Technology |
| PKI | Public Key Infrastructure |
| RSA | Rivest-Shamir-Adleman |
| UA | Unified Architecture |

4 Overview

4.1 General

The OPC Unified architecture multipart standard describes a number of *Services* and a variety of information models. These *Services* and information models can be referred to as features of a *Server* or *Client*. *Servers* and *Clients* need to be able to describe which features they support and wish to have certified. This document provides a grouping of these features. The individual features are grouped into *ConformanceUnits* which are further grouped into *Profiles*. Figure 1 provides an overview of the interactions between *Profiles*, *ConformanceUnits* and *TestCases*. The large arrows indicate the components that are used to construct the parent. For example a *Profile* is constructed from *Profiles* and *ConformanceUnits*. The figure also illustrates a feature of the OPC UA Compliance Test Tool (CTT), in that it will test if a requested *Profile* passes all *ConformanceUnits*. It will also test all other *ConformanceUnits* and report any other *Profiles* that pass conformance testing. The individual *TestCases* are defined in separate documents see Compliance Part 8 UA *Server* and Compliance Part 9 UA *Client*. The *TestCases* are related back to the appropriate *ConformanceUnits* defined in this standard. This relationship is also displayed by the OPC UA Compliance Test Tool.

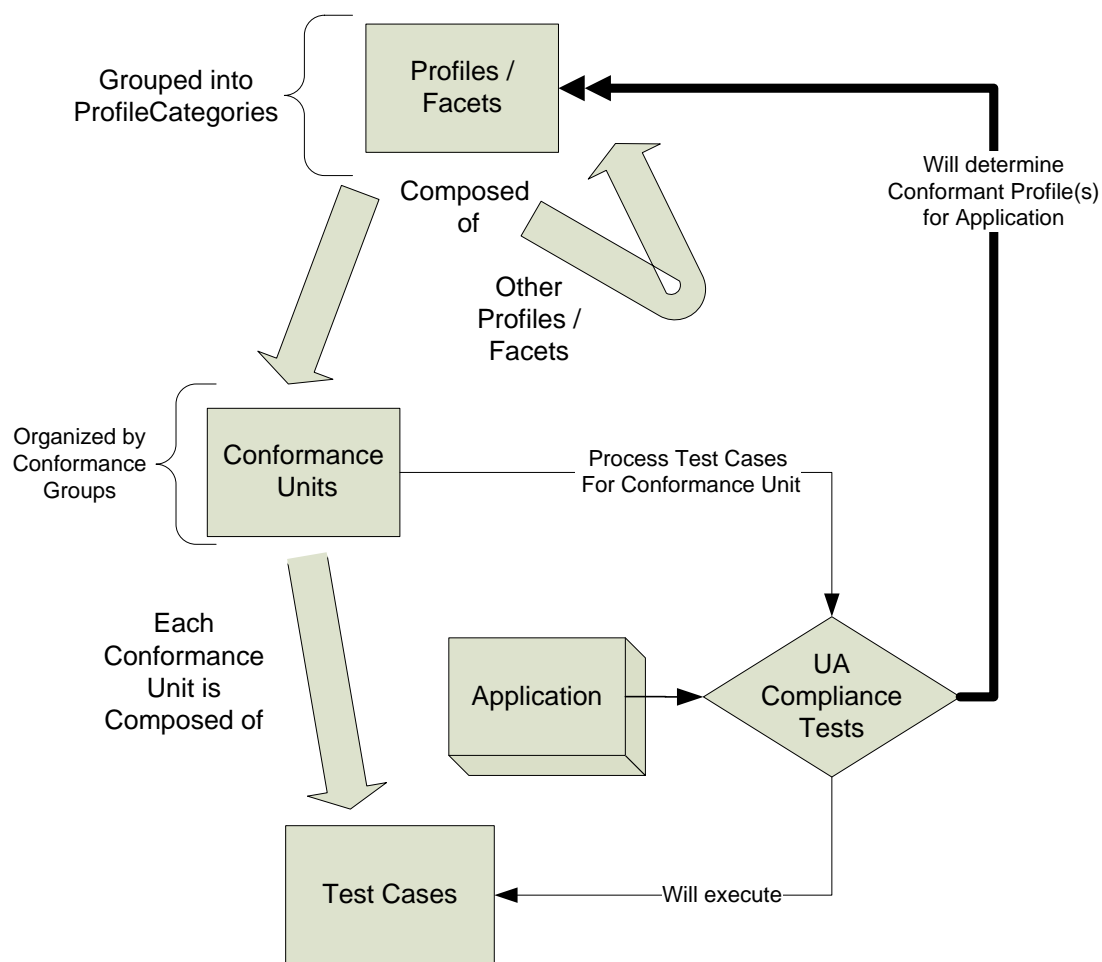


Figure 1 – Profile – ConformanceUnit – TestCases

4.2 ConformanceUnit

Each *ConformanceUnit* represents a specific set of features (e.g. a group of services, portions of services or information models) that can be tested as a single entity. *ConformanceUnits* are the building blocks of a *Profile*. Each *ConformanceUnit* can also be used as a test category. For each *ConformanceUnit*, there would be a number of *TestCases* that test the functionality described by the *ConformanceUnit*. The description of a *ConformanceUnit* is intended to provide enough information to illustrate the required functionality, but in many cases to obtain a complete understanding of the *ConformanceUnit* the reader may be required to also examine the appropriate part of OPC UA. Additional Information regarding testing of a *ConformanceUnit* are provided in the Compliance Part 8 UA Server or Compliance Part 9 UA Client test standards.

The same features do not appear in more than one *ConformanceUnit*.

4.3 Profiles

A *Profile* is a named aggregation of *ConformanceUnits* and other *Profiles*. To support a *Profile*, an application has to support the *ConformanceUnits* and all aggregated *Profiles*. The definition of *Profiles* is an ongoing activity, in that it is expected that new *Profiles* will be added in the future.

An OPC UA Application will typically support multiple *Profiles*.

Multiple *Profiles* may include the same *ConformanceUnit*.

Testing of a *Profile* consists of testing the individual *ConformanceUnits* that comprise the *Profile*.

Profiles are named based on naming conventions (see 6.3 for details).

4.4 Profile Categories

Profiles are grouped into categories to help vendors and end users understand the applicability of a *Profile*. A *Profile* can be assigned to more than one category.

Table 1 contains the list of currently defined *ProfileCategories*.

Table 1 – ProfileCategories

| Category | Description |
|-----------|--|
| Client | <i>Profiles</i> of this category specify functions of an OPC UA <i>Client</i> . |
| Security | <i>Profiles</i> of this category specify security related functions. Security policies are part of this category. The URI of security policies has to be part of an Endpoint Description returned from the GetEndpoints service. <i>Profiles</i> of this category apply to <i>Servers</i> and <i>Clients</i> . |
| Server | <i>Profiles</i> of this category specify functions of an OPC UA <i>Server</i> . The URI of such <i>Profiles</i> can be exposed in the <i>Server</i> capabilities. |
| Transport | <i>Profiles</i> of this category specify specific protocol mappings. The URI of such <i>Profiles</i> has to be part of an Endpoint Description. These <i>Profiles</i> apply to <i>Servers</i> and <i>Clients</i> . |

5 ConformanceUnits

5.1 Overview

A *ConformanceUnit* represents an individually testable entity. For improved clarity, the large list of *ConformanceUnits* is arranged into named *ConformanceGroups*. These groups reflect the *Service Sets* in Part 4 and the OPC UA information models. Table 2 lists the *ConformanceGroups*. These groups and the *ConformanceUnits* that they describe are detailed in the Subclauses of chapter 5 starting with clause 5.2 *ConformanceGroups* have no impact on testing; they are used only for organizational reasons, i.e. to simplify the readability of this document.

Table 2 – ConformanceGroups

| Group | Description |
|-----------------------|---|
| Address Space Model | Defines <i>ConformanceUnits</i> for various features of the OPC UA <i>AddressSpace</i> . |
| Aggregates | All <i>ConformanceUnits</i> that are related to <i>Aggregates</i> , including individual <i>ConformanceUnits</i> for each supported <i>Aggregate</i> as described in Part 13. |
| Alarms and Conditions | All <i>ConformanceUnits</i> that are associated with the OPC UA information model for <i>Conditions</i> , acknowledgeable <i>Conditions</i> , confirmations and <i>Alarms</i> as specified in Part 9. |
| Attribute Services | Includes <i>ConformanceUnits</i> to read or write current or historical <i>Attribute</i> values. |
| Auditing | User level security includes support for security audit trails, with traceability between <i>Client</i> and <i>Server</i> audit logs. |
| Base Information | All information elements as defined in Part 5. |
| Data Access | <i>ConformanceUnits</i> specific to <i>Clients</i> and <i>Servers</i> that deal with the representation and use of automation data as specified in Part 8. |
| Discovery Services | <i>ConformanceUnits</i> which focus on <i>Server Endpoint Discovery</i> . |
| Historical Access | Access to archived data of node <i>Attribute</i> values or <i>Events</i> . |
| Method Services | <i>Methods</i> represent the function calls of <i>Objects</i> . <i>Methods</i> are invoked and return only after completion (successful or unsuccessful). |
| Miscellaneous | This group contains <i>ConformanceUnits</i> that cover miscellaneous subjects, such as recommended behaviours, |

| Group | Description |
|--------------------------|--|
| | documentation etc. These <i>ConformanceUnits</i> typically do not fit into any of the other groups. |
| Monitored Item Services | <i>Clients</i> define <i>MonitoredItems</i> to subscribe to data and Events. Each <i>MonitoredItem</i> identifies the item to be monitored and the <i>Subscription</i> to use to send <i>Notifications</i> . |
| Node Management Services | Bundles <i>ConformanceUnits</i> for all <i>Services</i> to add and delete OPC UA <i>AddressSpace Nodes</i> and <i>References</i> . |
| Protocol and Encoding | Covers all transport and encoding combinations that are specified in Part 6. |
| Query Services | A Query may be used to provide advanced filtering and return a subset of data. |
| Redundancy | The design of OPC UA ensures that vendors can create redundant <i>Clients</i> and redundant <i>Servers</i> in a consistent manner. Redundancy may be used for high availability, fault tolerance and load balancing. |
| Security | Security related <i>ConformanceUnits</i> that can be profiled this includes all aspects of security. |
| Session Services | An (OPC UA) <i>Session</i> is an application layer connection. |
| Subscription Services | Subscriptions are used to report <i>Notifications</i> to the <i>Client</i> . |
| View Services | <i>Clients</i> use the View <i>Service Set</i> to navigate through the OPC UA <i>AddressSpace</i> or through a View (a subset) of the OPC UA <i>AddressSpace</i> . |

5.2 Services

Tables 3 to 10 describe *ConformanceUnits* for the *Services* specified in Part 4. The tables correlate with the *Service Sets*.

A single *ConformanceUnit* can reference several *Services* (e.g. CreateSession, ActivateSession and CloseSession) but can also refer to individual aspects of *Services* (e.g. the use of ActivateSession to impersonate a new user).

Each table includes a listing of the *Profile Category* to which a *ConformanceUnit* belongs, the title and description of the *ConformanceUnit*. In some cases, a *ConformanceUnit* will be derived from another *ConformanceUnit*. This parent unit will then be specified in the description of each derived unit. In such cases the derived units inherit all of the tests of its parent plus one or more additional TestCases. These TestCases can only further restrict the existing TestCases. An example would be one in which the number of connections is tested, where the TestCase of the parent required at least one connection and the derived *ConformanceUnit* would require a *TestCase* for at least five connections.

The *Discovery Service Set* is composed of multiple *ConformanceUnits* (see Table 3). All *Servers* provide some aspects of this functionality; see *Profiles* categorized as *Server Profiles* for details. *Clients* may support some aspects of this functionality; see *Profiles* categorized as *Client Profiles* for details.

Table 3 – Discovery Services

| Category | Title | Description |
|----------|-----------------------------|--|
| Server | Discovery Get Endpoints | Support the GetEndpoints <i>Service</i> to obtain all Endpoints of the <i>Server</i> . This includes filtering based on <i>Profiles</i> . |
| Server | Discovery Find Servers Self | Support the FindServers <i>Service</i> only for itself. |
| Server | Discovery Register | Call the RegisterServer <i>Service</i> to register itself (OPC UA <i>Server</i>) with an external <i>Discovery Service</i> via a secure channel with a <i>SecurityMode</i> other than "None". |
| Server | Discovery Register2 | Call the RegisterServer2 <i>Service</i> to register with an external <i>Discovery Service</i> via a <i>Secure Channel</i> with a |

| Category | Title | Description |
|----------|---|---|
| | | <p><i>SecurityMode</i> other than "None". This includes passing a list of short capability identifiers.</p> <p>The use of these identifiers is specified in Part 12; the complete list can be found in http://www.opcfoundation.org/UA/schemas/1.03/ServerCapabilityIdentifiers.csv.</p> |
| Server | Discovery Configuration | <p>Allow configuration of the <i>Discovery Server</i> URL where the <i>Server</i> will register itself.</p> <p>Allow complete disabling of registration with a <i>Discovery Server</i>.</p> |
| Server | Discovery Server Announcement using mDNS | <p>Provide mDNS functionality to announce a <i>Server</i> with its capabilities. The capability identifiers and the use of mDNS records for the purpose of OPC UA Discovery is specified in Part 12.</p> <p>Note that this functionality is only required for <i>Servers</i> that do not register with an LDS.</p> <p>The use of capability identifiers in mDNS records is specified in Part 12; the complete list can be found in http://www.opcfoundation.org/UA/schemas/1.03/ServerCapabilityIdentifiers.csv.</p> |
| Client | Discovery Client Find Servers Basic | Uses the <i>FindServers Service</i> to obtain all <i>Servers</i> installed on a given platform. |
| Client | Discovery Client Find Servers with URI | Use <i>FindServers Service</i> to obtain URLs for specific <i>Server</i> URIs. |
| Client | Discovery Client Find Servers Dynamic | Detect new <i>Servers</i> after an initial <i>FindServers Service</i> call. |
| Client | Discovery Client Find Servers on Network using LDS-ME | Use <i>FindServersOnNetwork Service</i> to obtain URLs for specific <i>Server</i> URIs. Note that this <i>Service</i> is available via the <i>Local Discovery Server</i> with multicast extension (LDS-ME). |
| Client | Discovery Client Find Servers on Network using mDNS | <p>Use mDNS based <i>Service Discovery</i> to locate <i>Servers</i> on the same multicast network. The contents of mDNS records for OPC UA Discovery are described in Part 12.</p> <p>Note that this functionality is only required for <i>Clients</i> when there is no <i>Local Discovery Server</i> with multicast extension (LDS-ME).</p> <p>The use of capability identifiers in mDNS records is specified in Part 12; the complete list can be found in http://www.opcfoundation.org/UA/schemas/1.03/ServerCapabilityIdentifiers.csv.</p> |
| Client | Discovery Client Find Servers on Network | Support one of the options to locate <i>Servers</i> on the network. |
| Client | Discovery Client Find Servers in GDS | Use the <i>QueryServers Method</i> on the GDS Directory Object to locate <i>Servers</i> that meet filter criteria specified in the request. This <i>Method</i> is specified in Part 12. |
| Client | Discovery Client Get Endpoints Basic | Uses the <i>GetEndpoints Service</i> to obtain all <i>Endpoints</i> for a given <i>Server</i> URI. |
| Client | Discovery Client Get Endpoints Dynamic | Detect changes to the <i>Endpoints</i> after an initial <i>GetEndpoints Service</i> call. |

| Category | Title | Description |
|----------|--|---|
| Client | Discovery Client Configure Endpoint | Allow specification of an Endpoint without going through the <i>Discovery Service Set</i> . |

The *Session Service Set* is composed of multiple *ConformanceUnits* (see Table 4). The *CreateSession*, *ActivateSession*, and *CloseSession* services are supported as a single unit. All *Servers* and *Clients* provide this functionality.

Table 4 – Session Services

| Category | Title | Description |
|----------|--|--|
| Server | Session General Service Behaviour | Implement basic <i>Service</i> behaviour. This includes in particular: <ul style="list-style-type: none"> – checking the authentication token – returning the requestHandle in responses – returning available diagnostic information as requested with the 'returnDiagnostics' parameter – respecting a timeoutHint |
| Server | Session Base | Support the <i>Session Service Set</i> (<i>CreateSession</i> , <i>ActivateSession</i> , <i>CloseSession</i>) except the use of <i>ActivateSession</i> to change the <i>Session</i> user. This includes correct handling of all parameters that are provided. Note that for the <i>CreateSession</i> and <i>ActivateSession</i> services, if the <i>SecurityMode</i> = <i>None</i> then: <ol style="list-style-type: none"> 1) The Application <i>Certificate</i> and <i>Nonce</i> are optional. 2) The signatures are null/empty. The details of this are described in Part 4. |
| Server | Session Change User | Support the use of <i>ActivateSession</i> to change the <i>Session</i> user. |
| Server | Session Cancel | Support the <i>Cancel Service</i> to cancel outstanding requests. |
| Server | Session Minimum 1 | Support minimum 1 <i>Session</i> (total). |
| Server | Session Minimum 2 Parallel | Support minimum 2 parallel <i>Sessions</i> (total for all <i>Clients</i>). |
| Server | Session Minimum 50 Parallel | Support minimum 50 parallel <i>Sessions</i> (total for all <i>Clients</i>). |
| Client | Session Client General Service Behaviour | Implement basic <i>Service</i> behaviour. This includes in particular: <ul style="list-style-type: none"> – including the proper authentication token of the <i>Session</i> – creating a requestHandle if needed – requesting diagnostic information with the 'returnDiagnostics' parameter – evaluate the <i>serviceResult</i> and operational results |
| Client | Session Client Base | Use the <i>Session Service Set</i> (<i>CreateSession</i> , <i>ActivateSession</i> , and <i>CloseSession</i>) except the use of <i>ActivateSession</i> to change the <i>Session</i> user. This includes correct handling of all parameters that are provided. Note that for the <i>CreateSession</i> and <i>ActivateSession</i> services, if the <i>SecurityMode</i> = <i>None</i> then: <ol style="list-style-type: none"> 1) The Application <i>Certificate</i> and <i>Nonce</i> are optional. 2) The signatures are null/empty. |
| Client | Session Client Multiple Connections | Support unlimited connections (client side) with multiple <i>Servers</i> . Any limit on numbers of connections is from server side. May have a memory based limit, but not a software constraint limit. |
| Client | Session Client Renew NodeIds | This <i>ConformanceUnit</i> applies to <i>Clients</i> that allow persisting <i>NodeIds</i> . |

| Category | Title | Description |
|----------|--------------------------------|--|
| | | Verify that the Namespace Table has not changed for NodeIds that the <i>Client</i> has persisted and is going to re-use beyond a <i>Session</i> lifetime. If changes occurred the <i>Client</i> has to recalculate the Namespace Indices of the respective NodeIds. |
| Client | Session Client Impersonate | Uses <i>ActivateSession</i> to change the <i>Session</i> user (impersonation). |
| Client | Session Client KeepAlive | Make periodic requests to keep the <i>Session</i> alive. |
| Client | Session Client Detect Shutdown | Read or monitor the <i>ServerStatus/State Variable</i> to recognize a potential shutdown of the <i>Server</i> and clean up resources. |
| Client | Session Client Cancel | Use the <i>Cancel Service</i> to cancel outstanding requests. |
| Client | Session Client Auto Reconnect | Automatic <i>Client</i> reconnect including: <ul style="list-style-type: none"> – <i>ActivateSession</i> with new <i>SecureChannel</i> if <i>SecureChannel</i> is no longer valid but <i>Session</i> is still valid – Creation of a new <i>Session</i> only if <i>Session</i> is no longer valid |
| Client | Client Entry-Level Support | The <i>Client</i> is able to interoperate with <i>Servers</i> with lowest level functionality. This includes the ability to operate with a single <i>Session</i> , a pre-knowledge of the OPC UA Types (the <i>Server</i> may not expose them in the <i>AddressSpace</i>), and the ability to use <i>Read</i> vs. <i>Subscriptions</i> for monitoring. There may be further restrictions provided by the <i>Server</i> via the <i>Server</i> capabilities. |
| Client | Session Client Single Session | The <i>Client</i> shall interoperate with <i>Servers</i> that only support one <i>Session</i> . |

The *Node Management Service Set* is composed of multiple *ConformanceUnits* (see Table 5). *Servers* may provide some aspects of this functionality; see *Profiles* categorized as *Server Profiles* for details. *Clients* may support some aspects of this functionality; see *Profiles* categorized as *Client Profiles* for details.

Table 5 – Node Management Services

| Category | Title | Description |
|----------|-----------------------------|---|
| Server | Node Management Add Node | Support the <i>AddNodes Service</i> to add one or more <i>Nodes</i> into the OPC UA <i>AddressSpace</i> . |
| Server | Node Management Delete Node | Support the <i>DeleteNodes Service</i> to delete one or more <i>Nodes</i> from the OPC UA <i>AddressSpace</i> . |
| Server | Node Management Add Ref | Support the <i>AddReferences Service</i> to add one or more <i>References</i> to one or more <i>Nodes</i> in the OPC UA <i>AddressSpace</i> . |
| Server | Node Management Delete Ref | Support the <i>DeleteReferences Service</i> to delete one or more <i>References</i> of a <i>Node</i> in the OPC UA <i>AddressSpace</i> . |
| Client | Node Management Client | Uses <i>Node Management Services</i> to add or delete <i>Nodes</i> and to add or delete <i>References</i> in <i>Server's</i> OPC UA <i>AddressSpace</i> . |

The *View Service Set* is composed of a multiple *ConformanceUnits* (see Table 6). All *Servers* support some aspects of this conformance group. *Clients* may support some aspects of this functionality; see *Profiles* categorized as *Client Profiles* for details.

Table 6 – View Services

| Category | Title | Description |
|----------|---|---|
| Server | View Basic | Support the View <i>Service</i> Set (Browse, BrowseNext). |
| Server | View TranslateBrowsePath | Support TranslateBrowsePathsToNodeIds <i>Service</i> . |
| Server | View RegisterNodes | Support the RegisterNodes and UnregisterNodes <i>Services</i> as a way to optimize access to repeatedly used <i>Nodes</i> in the <i>Server's</i> OPC UA <i>AddressSpace</i> . |
| Server | View Minimum Continuation Point 01 | Support minimum 1 continuation point per <i>Session</i> . |
| Server | View Minimum Continuation Point 05 | Support minimum 5 continuation points per <i>Session</i> . This number has to be supported for at least half of the minimum required sessions. |
| Client | View Client Basic Browse | Uses Browse and BrowseNext <i>Services</i> to navigate through the <i>Server's</i> OPC UA <i>AddressSpace</i> . Make use of the referenceTypeld and the nodeClassMask to specify the needed <i>References</i> . |
| Client | View Client Basic ResultSet Filtering | Makes use of the resultMask parameter to optimize the result set to be returned by the <i>Server</i> . |
| Client | View Client TranslateBrowsePath | Uses the TranslateBrowsePathsToNodeIds <i>Service</i> to identify the NodeIds for <i>Nodes</i> where a starting <i>Node</i> and a BrowsePath is known. Makes use of bulk operations rather than multiple calls whenever possible. |
| Client | View Client RegisterNodes | Uses the RegisterNodes <i>Service</i> to optimize access for <i>Nodes</i> that are used repeatedly. Use UnregisterNodes when <i>Nodes</i> are not used anymore. |
| Client | View Client Remote Nodes Browse | The <i>Client</i> can browse to nodes that have an extended NodeID that reference a <i>Server</i> different than the originating <i>Server</i> . This includes automatic connection to the remote <i>Server</i> . It is acceptable that the <i>Server</i> configuration information be pre-configured on the <i>Client</i> and / or that the user is prompted to connect. |
| Client | View Client Remote Nodes Translate Browse | The <i>Client</i> can translate browse paths that include nodes with extended NodeID that reference a <i>Server</i> different than the originating <i>Server</i> and return them as part of the TranslateBrowsePathsToNodeIds <i>Service</i> . It is acceptable that the <i>Server</i> configuration information be pre-configured on the <i>Client</i> . |

The *Attribute Service* Set is composed of multiple *ConformanceUnits* (see Table 7). The majority of the *Attribute* service set is a core functionality of OPC UA and as such is supported by most *Servers*. Most *Clients* will also support some aspects of the *Attribute Service* Set

Table 7 – Attribute Services

| Category | Title | Description |
|----------|-------------------------|---|
| Server | Attribute Read | Supports the Read <i>Service</i> to read one or more <i>Attributes</i> of one or more <i>Nodes</i> . This includes support of the IndexRange parameter to read a single element or a range of elements when the <i>Attribute</i> value is an array. |
| Server | Attribute Read Complex | Supports reading and encoding Values with Structured DataTypes. |
| Server | Attribute Write Values | Supports writing to values to one or more <i>Attributes</i> of one or more <i>Nodes</i> . |
| Server | Attribute Write Complex | Supports writing and decoding Values with Structured DataTypes. |

| Category | Title | Description |
|----------|--|---|
| Server | Attribute Write StatusCode & Timestamp | Supports writing of StatusCode and Timestamps along with the Value. |
| Server | Attribute Write Index | Supports the IndexRange to write a single element or a range of elements when the <i>Attribute</i> value is an array. |
| Server | Attribute Alternate Encoding | Supports alternate Data Encoding when reading value <i>Attributes</i> . By default, every <i>Server</i> has to support the Data Encoding of the currently used <i>Stack Profile</i> (i.e. binary with UA Binary Encoding and XML with XML Encoding). This <i>ConformanceUnit</i> – when supported – specifies that the other Data Encoding is supported in addition. |
| Server | Attribute Historical Read | Supports the HistoryRead <i>Service</i> . The details of what aspects of this service are used are listed in additional <i>ConformanceUnits</i> , but at least one of ReadRaw, ReadProcessed, ReadModified, ReadAtTime or ReadEvents must be supported. |
| Server | Attribute Historical Update | Supports the HistoryUpdate service. The details of the supported features of this service are described by additional <i>ConformanceUnits</i> , but at least one of the following must be supported: InsertData, InsertEvents, ReplaceData, ReplaceEvents, UpdateData, UpdateEvents, DeleteData, DeleteEvents or DeleteAtTime. |
| Client | Attribute Client Read Base | Use the Read <i>Service</i> to read one or more <i>Attributes</i> of one or more <i>Nodes</i> . This includes use of an IndexRange to select a single element or a range of elements when the <i>Attribute</i> value is an array. <i>Clients</i> shall use bulk operations whenever possible to reduce the number of <i>Service</i> invocations. |
| Client | Attribute Client Remote Nodes Attribute Access | The <i>Client</i> can retrieve attributes of nodes that have an extended NodeID that reference a <i>Server</i> different than the originating <i>Server</i> . This requires a connection to the remote <i>Server</i> for access (not necessarily displayed as a connection). It is acceptable that the <i>Server</i> configuration information be pre-configured on the <i>Client</i> . |
| Client | Attribute Client Read with proper Encoding | This <i>ConformanceUnit</i> refers to the ability of a <i>Server</i> to support more than one Data Encoding for <i>Attribute</i> values. <i>Clients</i> can discover the available encodings and can explicitly choose one when calling the Read <i>Service</i> . |
| Client | Attribute Client Read Complex | Read and decode Values with Structured DataTypes. |
| Client | Attribute Client Write Base | Use the Write <i>Service</i> to write values to one or more <i>Attributes</i> of one or more <i>Nodes</i> . This includes use of an IndexRange to select a single element or a range of elements when the <i>Attribute</i> value is an array. <i>Clients</i> shall use bulk operations whenever possible to reduce the number of <i>Service</i> invocations. |
| Client | Attribute Client Write Complex | Write and Encode Values with Structured DataTypes. |
| Client | Attribute Client Write Quality & TimeStamp | Use the Write <i>Service</i> to also write StatusCode and/or Timestamps along with a Value. |
| Client | Attribute Client Historical Read | The <i>Client</i> makes use of the HistoryRead service. The details of which aspect of this service are used are provided by additional <i>ConformanceUnits</i> , but at least one or more of the following is used ReadRaw, |

| Category | Title | Description |
|----------|--|--|
| | | ReadAtTime, ReadProcessed, ReadModified or ReadEvents. |
| Client | Attribute Client Historical Updates | The <i>Client</i> makes use of the HistoryUpdate service. The details of this usage are provided by additional <i>ConformanceUnits</i> , but at least one or more of the following must be provided: InsertData, InsertEvent, ReplaceData, ReplaceEvent, UpdateData, UpdateEvents, DeleteData or DeleteEvents or DeleteAtTime. |

The *Method Service Set* is composed of *ConformanceUnits* (see Table 8). The primary *ConformanceUnits* provide support for the call functionality. *Servers* may provide some aspects of this functionality; see *Profiles* categorized as *Server Profiles* for details. *Clients* may support some aspects of this functionality; see *Profiles* categorized as *Client Profiles* for details.

Table 8 – Method Services

| Category | Title | Description |
|----------|-----------------------|--|
| Server | Method Call | Support the Call <i>Service</i> to call (invoke) a <i>Method</i> which includes support for <i>Method Parameters</i> . |
| Client | Method Client Call | Use the Call <i>Service</i> to call one or several <i>Methods</i> . |

The *MonitoredItem Service Set* is composed of multiple *ConformanceUnits* (see Table 9). *Servers* may provide some aspects of this functionality; see *Profiles* categorized as *Server Profiles* for details. *Clients* may support some aspects of this functionality; see *Profiles* categorized as *Client Profiles* for details.

Table 9 – Monitored Item Services

| Category | Title | Description |
|----------|------------------------------------|--|
| Server | Monitor Basic | Support the following <i>MonitoredItem Services</i> : CreateMonitoredItems, ModifyMonitoredItems, DeleteMonitoredItems and SetMonitoringMode. |
| Server | Monitor Value Change | Support creation of <i>MonitoredItems</i> for <i>Attribute</i> value changes. This includes support of the <i>IndexRange</i> to select a single element or a range of elements when the <i>Attribute</i> value is an array. |
| Server | Monitored Items Deadband Filter | Supports an absolute Deadband filter as a <i>DataChangeFilter</i> for numeric data types. |
| Server | Monitor Aggregate Filter | Support for Aggregate filters for <i>MonitoredItems</i> . The result of this <i>ConformanceUnit</i> includes a list of Aggregates that are supported as part of the <i>Profile Certificate</i> . |
| Server | Monitor Alternate Encoding | Support alternate encoding when monitoring value <i>Attributes</i> . By default, every <i>Server</i> has to support the encoding of the currently used <i>Stack Profile</i> (i.e. binary with UA Binary Encoding and XML with XML Encoding). This <i>ConformanceUnit</i> – when supported – specifies that the other encoding is supported in addition. |
| Server | Monitor Items 2 | Support at least 2 <i>MonitoredItems</i> per <i>Subscription</i> where the size of each <i>MonitoredItem</i> is at least equal to size of Double. |
| Server | Monitor Items 10 | Support at least 10 <i>MonitoredItems</i> per <i>Subscription</i> where the size of each <i>MonitoredItem</i> is at least equal to size of Double. |
| Server | Monitor Items 100 | Support at least 100 <i>MonitoredItems</i> per <i>Subscription</i> . |

| Category | Title | Description |
|----------|-------------------------------------|---|
| | | This number has to be supported for at least half of the required Subscriptions for half of the required Sessions. |
| Server | Monitor Items 500 | Support at least 500 <i>MonitoredItems</i> per <i>Subscription</i> . This number has to be supported for at least half of the required Subscriptions for half of the required Sessions. |
| Server | Monitor QueueSize_1 | This <i>ConformanceUnit</i> does not require queuing when multiple value changes occur during a "publish period". I.e. the latest change will be sent in the <i>Notification</i> . |
| Server | Monitor MinQueueSize_02 | Support at least 2 queue entries for <i>MonitoredItems</i> . Servers often will adapt the queue size to the number of currently <i>MonitoredItems</i> . However, it is expected that Servers support this minimum queue size for at least one third of the supported <i>MonitoredItems</i> . |
| Server | Monitor MinQueueSize_05 | Support at least 5 queue entries for <i>MonitoredItems</i> . Servers often will adapt the queue size to the number of currently <i>MonitoredItems</i> . However, it is expected that Servers support this minimum queue size for at least one third of the supported <i>MonitoredItems</i> . |
| Server | Monitor QueueSize_ServerMax | This <i>ConformanceUnit</i> is for events. When the Client requests queueSize=MAXUInt32 the Server is to return the maximum queue size that it can support for event notifications as the revisedQueueSize. |
| Server | Monitor Triggering | Support the SetTriggering <i>Service</i> to create and/or delete triggering links for a triggering item. |
| Server | Monitor Events | Support creation of <i>MonitoredItems</i> for an "EventNotifier Attribute" for the purpose of <i>Event Notification</i> . The subscription includes supporting a filter that includes SimpleAttribute Operands and a select list of Operators. The list of Operators includes: Equals, IsNull, GreaterThan, LessThan, GreaterThanorEqual, LessThanorEqual, Like, Not, Between, InList, And, Or, Cast, BitwiseAnd, BitwiseOr. |
| Server | Monitor Complex Event Filter | Support for the 'TypeOf' complex Event filter operator. |
| Client | Monitor Client Value Change | Use the <i>MonitoredItem Service</i> Set to register items for changes in <i>Attribute</i> value. Use <i>CreateMonitoredItems</i> to register the <i>Node/Attribute</i> tuple. Set proper sampling interval, Deadband filter and queuing mode. Use disabling / enabling instead of deleting and re-creating a <i>MonitoredItem</i> . Use bulk operations rather than individual service requests to reduce communication overhead. |
| Client | Monitor Client Deadband Filter | Uses Absolute Deadband filters for subscriptions. |
| Client | Monitor Client by Index | Use the <i>IndexRange</i> to select a single element or a range of elements when the <i>Attribute</i> value is an array. |
| Client | Monitor Client Aggregate Filter | Uses Aggregate filters for Subscriptions. |
| Client | Monitor Client Events | Use the <i>MonitoredItem Service</i> Set to create <i>MonitoredItems</i> for <i>Event</i> notifications. |
| Client | Monitor Client Event Filter | Use the <i>Event</i> filter when calling <i>CreateMonitoredItems</i> to filter the desired Events and to select the columns to be provided for each <i>Event Notification</i> . |
| Client | Monitor Client Complex Event Filter | Use of the 'TypeOf' complex Event filter operator. |
| Client | Monitor Client Modify | Use <i>ModifyMonitoredItems Service</i> to change the configuration setting. |

| Category | Title | Description |
|----------|------------------------|---|
| | | Use SetMonitoringMode Service to disable / enable sampling and / or publishing. |
| Client | Monitor Client Trigger | Use the Triggering Model if certain items are to be reported only if some other item triggers. Use proper monitoring mode for these items. Use SetTriggering Service to link these items to the trigger item. |

The *Subscription Service Set* is composed of multiple *ConformanceUnits* (see Table 10). *Servers* may provide some aspects of this functionality; see *Profiles* categorized as *Server Profiles* for details. *Clients* may support some aspects of this functionality; see *Profiles* categorized as *Client Profiles* for details.

Table 10 – Subscription Services

| Category | Title | Description |
|----------|-------------------------------------|---|
| Server | Subscription Basic | Support the following <i>Subscription Services</i> : CreateSubscription, ModifySubscription, DeleteSubscriptions, Publish, Republish and SetPublishingMode. |
| Server | Subscription Minimum 1 | Support at least 1 Subscriptions per Session. This number has to be supported for all of the minimum required sessions. |
| Server | Subscription Minimum 02 | Support at least 2 Subscriptions per Session. This number has to be supported for at least half of the minimum required sessions. |
| Server | Subscription Minimum 05 | Support at least 5 Subscriptions per Session. This number has to be supported for at least half of the minimum required sessions. |
| Server | Subscription Publish Min 02 | Support at least 2 Publish Service requests per Session. This number has to be supported for all of the minimum required sessions. Support of a NotificationMessage retransmission queue is not required; if not available the Republish Service returns Bad_MessageNotAvailable. |
| Server | Subscription Publish Min 05 | Support at least 5 Publish Service requests per Session. This number has to be supported for at least half of the minimum required sessions. Support, as a minimum, the number of Publish requests per session as the size of the NotificationMessage retransmission queue for Republish. |
| Server | Subscription Publish Min 10 | Support at least 10 Publish Service requests per Session. This number has to be supported for at least half of the minimum required sessions. Support as a minimum, the number of Publish requests per session as the size of the NotificationMessage retransmission queue for Republish. |
| Server | Subscription Publish Discard Policy | Respect the specified policy for discarding Publish Service requests. If the maximum number of Publish Service requests has been queued and a new Publish Service request arrives, the "oldest" Publish request has to be discarded by returning the proper error. |
| Server | Subscription Transfer | Support TransferSubscriptions Service to transfer a Subscription from one Session to another. |

| Category | Title | Description |
|----------|------------------------------------|--|
| Server | Subscription Durable | Support setting <i>Subscriptions</i> in durable mode. This mode requires that collected data and events are stored and delivered even if a <i>Client</i> was disconnected for a longer time or the <i>Server</i> was restarted. |
| Client | Subscription Client Basic | Use the <i>Subscription</i> and <i>MonitoredItem Service</i> Set as an efficient means to detect changes of <i>Attribute</i> values and / or to receive <i>Event</i> occurrences. Set appropriate intervals for publishing, keep alive notifications and total <i>Subscription</i> lifetime. Supply a sufficient number of Publish requests to the <i>Server</i> so that <i>Notifications</i> can be sent whenever a publish timer expires. Acknowledge received <i>Notifications</i> with subsequent Publish requests. |
| Client | Subscription Fallback | The <i>Client</i> shall interoperate with <i>Servers</i> that do not support <i>Subscriptions</i> , or have exhausted <i>Subscription</i> limits, for Monitoring by using Read <i>Service</i> . |
| Client | Subscription Republish | Evaluate the sequence number in <i>Notifications</i> to detect lost <i>Notifications</i> . Use Republish to request missing <i>Notifications</i> . |
| Client | Subscription Modify | Allow modification of the <i>Subscription</i> configuration using the ModifySubscription <i>Service</i> . |
| Client | Subscription TransferSubscriptions | The <i>Client</i> supports transferring <i>Subscription</i> from other <i>Clients</i> . This <i>ConformanceUnit</i> is used as part of redundant <i>Clients</i> . |
| Client | Subscription Multiple | Use multiple <i>Subscriptions</i> to reduce the payload of individual <i>Notifications</i> . |
| Client | Subscription Publish Configurable | Send multiple Publish <i>Service</i> requests to assure that the <i>Server</i> is always able to send <i>Notifications</i> . The number of parallel Publish <i>Service</i> requests per <i>Session</i> shall be configurable. |
| Client | Subscription Durable | Use durable <i>Subscriptions</i> . |

5.3 Transport and communication related features

Table 11 describes security related *ConformanceUnits*. All of these *ConformanceUnits* apply equally to both *Clients* and *Servers*, where a *Client* uses the related security unit and a *Server* supports the use of it. These items are defined in detail in Part 6. It is recommended that a *Server* and *Client* support as many of these options as possible in order to achieve increased levels of interoperability. It is the task of an administrator to determine which of these *ConformanceUnits* are exposed in a given deployed *Server* or *Client* application.

Table 11 – Security

| Category | Title | Description |
|----------|---------------------------------|---|
| Security | Security Certificate Validation | A certificate will be validated as specified in Part 4. This includes among others structure and signature examination. Allowing for some validation errors to be suppressed by administration directive. |
| Security | Security None | A suite of algorithms that does NOT provide any security settings: -> SymmetricSignatureAlgorithm – Not Used -> SymmetricEncryptionAlgorithm – Not Used -> AsymmetricSignatureAlgorithm – Not Used -> SymmetricKeyWrapAlgorithm – Not Used -> AsymmetricEncryptionAlgorithm – Not Used |

| Category | Title | Description |
|----------|---|---|
| | | <p>-> KeyDerivationAlgorithm – Not Used</p> <p>-> DerivedSignatureKeyLength – 0</p> <p>The use of this suite of algorithms must be able to be enabled or disabled by an administrator.</p> |
| Security | Security None CreateSession ActivateSession | When SecurityPolicy=None, the CreateSession and ActivateSession service allow for a NULL/empty signature and do not require Application <i>Certificates</i> or a Nonce. |
| Security | Security None CreateSession ActivateSession 1.0 | The Client can connect to Servers that require a certificate being passed on Session establishment. The Client in this case will first try without a certificate and if this fails present a certificate. |
| Security | Security Basic 128Rsa15 | <p>A suite of algorithms that uses RSA15 as Key-Wrap-algorithm and 128-Bit for encryption algorithms.</p> <p>-> SymmetricSignatureAlgorithm – HmacSha1 – (http://www.w3.org/2000/09/xmlsig#hmac-sha1).</p> <p>-> SymmetricEncryptionAlgorithm – Aes128 – (http://www.w3.org/2001/04/xmlenc#aes128-cbc).</p> <p>-> AsymmetricSignatureAlgorithm – RsaSha1 – (http://www.w3.org/2000/09/xmlsig#rsa-sha1).</p> <p>-> AsymmetricKeyWrapAlgorithm – KwRsa15 – (http://www.w3.org/2001/04/xmlenc#rsa-1_5).</p> <p>-> AsymmetricEncryptionAlgorithm – Rsa15 – (http://www.w3.org/2001/04/xmlenc#rsa-1_5).</p> <p>-> KeyDerivationAlgorithm – PSha1 – (http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/dk/p_sha1).</p> <p>-> DerivedSignatureKeyLength – 128.</p> <p>-> MinAsymmetricKeyLength – 1024</p> <p>-> MaxAsymmetricKeyLength – 2048</p> <p>-> CertificateSignatureAlgorithm – Sha1</p> <p>If a certificate or any certificate in the chain is not signed with a hash that is Sha1 or stronger then the certificate shall be rejected.</p> |
| Security | Security Basic 256 | <p>A suite of algorithms that are for 256-Bit encryption, algorithms include:</p> <p>-> SymmetricSignatureAlgorithm – HmacSha1 – (http://www.w3.org/2000/09/xmlsig#hmac-sha1).</p> <p>-> SymmetricEncryptionAlgorithm – Aes256 – (http://www.w3.org/2001/04/xmlenc#aes256-cbc).</p> <p>-> AsymmetricSignatureAlgorithm – RsaSha1 – (http://www.w3.org/2000/09/xmlsig#rsa-sha1).</p> <p>-> AsymmetricKeyWrapAlgorithm – KwRsaOaep – (http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p).</p> <p>-> AsymmetricEncryptionAlgorithm – RsaOaep – (http://www.w3.org/2001/04/xmlenc#rsa-oaep).</p> <p>-> KeyDerivationAlgorithm – PSha1 – (http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/dk/p_sha1).</p> <p>-> DerivedSignatureKeyLength – 192.</p> <p>-> MinAsymmetricKeyLength – 1024</p> <p>-> MaxAsymmetricKeyLength – 2048</p> <p>-> CertificateSignatureAlgorithm – Sha1 [deprecated] or Sha256 [recommended]</p> <p>If a certificate or any certificate in the chain is not signed with a hash that is Sha1 or stronger then the certificate shall be rejected.</p> |

| Category | Title | Description |
|----------|--|--|
| | | Both Sha1 and Sha256 shall be supported. However, it is recommended to use Sha256 since Sha1 is considered not secure anymore. |
| Security | Security Basic 256 Sha256 | <p>A suite of algorithms that are for 256-Bit encryption, algorithms include.</p> <ul style="list-style-type: none"> -> SymmetricSignatureAlgorithm – Hmac_Sha256 – (http://www.w3.org/2000/09/xmlsig#hmac-sha256). -> SymmetricEncryptionAlgorithm – Aes256_CBC – (http://www.w3.org/2001/04/xmlenc#aes256-cbc). -> AsymmetricSignatureAlgorithm – Rsa_Sha256 – (http://www.w3.org/2001/04/xmlsig#rsa-sha256). -> AsymmetricKeyWrapAlgorithm – KwRsaOaep – (http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p). -> AsymmetricEncryptionAlgorithm – Rsa_Oaep – (http://www.w3.org/2001/04/xmlenc#rsa-oaep). -> KeyDerivationAlgorithm – PSHA256 – (http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/dk/p_sha256). -> DerivedSignatureKeyLength – 256 -> MinAsymmetricKeyLength – 2048 -> MaxAsymmetricKeyLength – 4096 -> CertificateSignatureAlgorithm – Sha256 <p>If a certificate or any certificate in the chain is not signed with a hash that is Sha256 or stronger then the certificate shall be rejected.</p> <p>Support for this security profile may require support for a second application instance certificate, with a larger keysize. Applications shall support multiple Application Instance <i>Certificates</i> if required by supported Security Policies and use the certificate that is required for a given security endpoint.</p> |
| Security | Security General TLS | This <i>ConformanceUnit</i> indicates that at least one of the transport security <i>Profiles</i> for TLS is supported by this application. It is used in TLS transport <i>Profiles</i> , but the choice of transport security profile is optional. The actual used security profile will default to the most secure one. |
| Security | Security TLS_RSA_WITH_AES_256_CBC_SHA256 | The connection is established using TLS_RSA_WITH_AES_256_CBC_SHA256. That has a MinAsymmetricKeyLength – 2048, MaxAsymmetricKeyLength – 4096, AsymmetricSignatureAlgorithm – RSA_SHA256. (TLS 1.2) |
| Security | Security TLS_DHE_RSA_WITH_AES_nnn_CBC_SHA256 | The connection is established using TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 or TLS_DHE_RSA_WITH_AES_256_CBC_SHA256. That has a MinAsymmetricKeyLength – 2048, MaxAsymmetricKeyLength – 4096, CertificateSignatureAlgorithm – RSA_SHA256. (TLS 1.2). Clients and Servers have to support both algorithms. |
| Security | Security Encryption Required | Encryption is required using the algorithms provide in the security algorithm suite. |
| Security | Security Signing Required | Signing is required using the algorithms provide in the security algorithm suite. |
| Security | Security Time Synch – Configuration | Application supports configuring acceptable clock skew. |
| Security | Security Time Synch – NTP / OS Based support | Application supports time synchronization, either via an implementation of Network Time Protocol (NTP), or via features of a standard operating system. |

| Category | Title | Description |
|----------|--|--|
| Security | Security Time Synch – UA based support | An application makes use of the responses header timestamp provided by a configured well know source, such as a <i>Discovery Server</i> to synchronize the time on the application and that this time synchronization occurs periodically. Use of this <i>TimeSyncing</i> can be configured. |
| Security | Security Administration | Allow configuration of the following Security related items. <ul style="list-style-type: none"> * select the allowed User identification policy or policies (User Name/Password or X509 or Kerberos or Anonymous). * enable/disable the security policy "None" or other security policies. * enable/disable endpoints with <i>MessageSecurityMode</i> SIGN or SIGNANDENCRYPT. * set the permitted certification authorities. * define how to react to unknown <i>Certificates</i>. * allow accepting any valid <i>Certificate</i> |
| Security | Security Administration – XML Schema | Support the OPC UA defined XML schema for importing and exporting security configuration information. This schema is defined in Part 6. |
| Security | Security Certificate Administration | Allow a site administrator to be able to assign a site specific <i>ApplicationInstanceCertificate</i> and if desired to configure a site specific <i>Certificate Authority (CA)</i> . |
| Security | Security Default ApplicationInstance Certificate | An application, when installed, has a default <i>ApplicationInstanceCertificate</i> that is valid. The default <i>ApplicationInstanceCertificate</i> shall either be created as part of the installation or installation instructions explicitly describe the process to create and apply a default <i>ApplicationInstanceCertificate</i> to the application. |
| Security | Security – No Application Authentication | The <i>Server</i> supports being able to be configured for no application authentication, just User authentication and normal encryption/signing: <ul style="list-style-type: none"> - Configure <i>Server</i> to accept all certificates - <i>Certificates</i> are just used for message security (signing and encryption) - Users level is used for authentication |
| Security | Best Practice – Audit Events | Subscriptions for Audit Events are restricted to authorized personnel. A <i>Server</i> may also reject a Subscription for Audit Events that is not over a Secure Channel if one is available. |
| Security | Best Practice – Alarm Handling | A <i>Server</i> should restrict critical alarm functionality to users that have the appropriate rights to perform these actions. This would include disabling or alarms, shelving of alarms and generation of dialog messages. It would also include other security related functionality such maintaining appropriate timeouts for shelving and dialogs and preventing an overload of dialog messages. |
| Security | Best Practice – Random Numbers | All random numbers that are required for security use appropriate cryptographic library based random number generators. |
| Security | Best Practice – Timeouts | The user is able to configure reasonable timeouts for Secure Channels, Sessions and Subscriptions to limit denial of service and resource consumption issues (see Part 2 for additional details). |
| Security | Best Practice – Administrative Access | The <i>Server</i> and <i>Client</i> allow for appropriate restriction of access to administrative personnel. This includes multiple levels of administrative access on platforms that support multiple administrative roles (such as Windows or Linux). |

| Category | Title | Description |
|----------|--|--|
| Security | Best Practice – Strict Message Handling | The application assures that messages that are illegally or incorrectly formed are rejected with appropriate error codes or appropriate actions as specified in Part 4 and Part 6. |
| Security | Best Practice – Audit Events Client | Audit tracking system connects to a Server using a Secure Channel and under the appropriate administrative rights to allow access to Audit Events. |
| Security | Security User Name Password | <p>The Server supports User Name/Password combination(s).</p> <p>The token will be encrypted if required by the security policy of the User Token Policy or by the security policy of the endpoint. An unencrypted token either requires message encryption or means outside the scope of OPC UA to secure the identity token so that it cannot be retrieved by sniffing the communication. One option would be a secure transport like a VPN.</p> |
| Security | Security User X509 | The Server supports a public/private key pair for user identity. The use of this feature must be able to be enabled or disabled by an administrator. |
| Security | Security User IssuedToken Kerberos | <p>The Server supports a Kerberos Server token for User Identity. The use of this feature must be able to be enabled or disabled by an Administrator. The use of this token is defined in Kerberos Token Documentation.</p> <p>The token will be encrypted if required by the security policy of the User Token Policy or by the security policy of the endpoint. An unencrypted token either requires message encryption or means outside the scope of OPC UA to secure the identity token so that it cannot be retrieved by sniffing the communication. One option would be a secure transport like a VPN.</p> |
| Security | Security User IssuedToken Kerberos Windows | <p>The Server supports the Windows implementation of Kerberos Tokens. This <i>ConformanceUnit</i> only applies if the "Security User IssuedToken Kerberos" is supported.</p> <p>The token will be encrypted if required by the security policy of the User Token Policy or by the security policy of the endpoint. An unencrypted token either requires message encryption or means outside the scope of OPC UA to secure the identity token so that it cannot be retrieved by sniffing the communication. One option would be a secure transport like a VPN.</p> |
| Security | Security User Anonymous | The Server provides support for Anonymous access. The use of this feature must be able to be enabled or disabled by an Administrator. By default Anonymous access shall be disabled. |
| Security | Security User IssuedToken Kerberos Client | <p>A Client uses a Kerberos Server token. The use of this token is defined by the Kerberos documentation.</p> <p>The token will be encrypted if required by the security policy of the User Token Policy or by the security policy of the endpoint. An unencrypted token either requires message encryption or means outside the scope of OPC UA to secure the identity token so that it cannot be retrieved by sniffing the communication. One option would be a secure transport like a VPN.</p> |

| Category | Title | Description |
|----------|--|--|
| Security | Security User IssuedToken Kerberos Windows Client | <p>A <i>Client</i> uses the Windows implementation of Kerberos tokens. This <i>ConformanceUnit</i> only applies if the "Security User IssuedToken Kerberos <i>Client</i>" is supported.</p> <p>The token will be encrypted if required by the security policy of the User Token Policy or by the security policy of the endpoint. An unencrypted token either requires message encryption or means outside the scope of OPC UA to secure the identity token so that it cannot be retrieved by sniffing the communication. One option would be a secure transport like a VPN.</p> |
| Security | Security User Name Password Client | <p>A <i>Client</i> uses a User Name/Password combination.</p> <p>The token will be encrypted if required by the security policy of the User Token Policy or by the security policy of the endpoint. An unencrypted token either requires message encryption or means outside the scope of OPC UA to secure the identity token so that it cannot be retrieved by sniffing the communication. One option would be a secure transport like a VPN.</p> |
| Security | Security User X509 Client | A <i>Client</i> uses a public/private key pair for user identity. This includes all validation and trust issues associated with a certificate. |
| Security | Pull Model for Global Certificate and TrustList Management | Support the <i>Certificate Management Services</i> of UA Part 12 for the Pull model to manage <i>Application Instance Certificates</i> and <i>Trust Lists</i> including <i>Revocation Lists</i> . |
| Security | Push Model for Global Certificate and TrustList Management | Support the <i>Certificate Management Services</i> of UA Part 12 for the Push model to manage <i>Application Instance Certificates</i> and <i>Trust Lists</i> including <i>Revocation Lists</i> . |
| Security | Pull or Push Model | Support the <i>Certificate Management Services</i> of UA Part 12 to manage <i>Application Instance Certificates</i> and <i>Trust Lists</i> including <i>Revocation Lists</i> . Either Pull or Push model shall be supported. |

Table 12 describes protocol and encoding related features that can be profiled. These features are defined in detail in Part 6. It is recommended that *Servers* and *Clients* support as many of these options as possible for greatest interoperability.

Table 12 – Protocol and Encoding

| Category | Title | Description |
|-----------|---------------------------------|--|
| Server | Protocol Configuration | Allow administration of the Endpoints and the port number used by the Endpoints. |
| Transport | Protocol TCP Binary UA Security | Support the UA TCP transport protocol with UA Binary Encoding and with UA Secure Conversation. |
| Transport | Protocol HTTPS with UA Binary | Support the HTTPS protocol with UA Binary Encoding. |
| Transport | Protocol HTTPS with Soap | Support the HTTPS protocol with Soap-based Xml Encoding. |
| Transport | Protocol Soap Xml WS Security | Support "SOAP/HTTP" transport with XML Encoding and with WS Secure Conversation. |

| Category | Title | Description |
|-----------|--|--|
| Transport | Protocol Soap Binary WS Security | Support "SOAP/HTTP" transport with UA Binary Encoding and with WS Secure Conversation. |

5.4 Information Model and AddressSpace related features

Table 13 describes Base features related items that can be profiled. For additional information about these items, please refer to Part 3, Part 5 and Part 6. *Servers* with a larger resource capacity would support most of this functionality, but smaller resource constraint *Server* may omit some of this functionality. Many *Clients* would utilize some of this functionality and more robust *Clients* would utilize most of this functionality.

Table 13 – Base information

| Category | Title | Description |
|----------|------------------------------------|---|
| Server | Base Info Core Structure | The <i>Server</i> supports the <i>Server Object</i> , <i>ServerCapabilities</i> and supports the OPC UA <i>AddressSpace</i> structure. |
| Server | Base Info Server Capabilities | The <i>Server</i> supports publishing of the <i>Server</i> limitation in the <i>ServerCapabilities</i> , including <i>MaxArrayLength</i> , <i>MaxStringLength</i> , <i>MaxNodePerRead</i> , <i>MaxNodesPerWrite</i> , <i>MaxNodesPerSubscription</i> and <i>MaxNodesPerBrowse</i> . |
| Server | Base Info Progress Events | The <i>Server</i> exposes if generation of Progress events for long running service calls such as <i>HistoryRead</i> or <i>Query</i> is supported. If it is listed as supported in <i>ServerCapabilities</i> , then the actual events are verified. |
| Server | Base Info Diagnostics | The <i>Server</i> supports the collection of diagnostic information. The <i>EnabledFlag</i> in the <i>ServerDiagnostics</i> Object can be set TRUE and in that case all static and dynamic Objects and Variables for diagnostic data as defined in UA Part 5 are supported. |
| Server | Base Info System Status | The <i>Server</i> supports generating <i>SystemStatusChangeEvent</i> indicating shutdown of the <i>Server</i> (<i>SourceNode=Server</i>). |
| Server | Base Info System Underlying System | The <i>Server</i> supports generating <i>SystemStatusChangeEvent</i> indicating changes to an Underlying System (<i>SourceNode = Server</i>). This event can also be used to indicate that the OPC UA <i>Server</i> has underlying systems. |
| Server | Base Info Device Failure | The <i>Server</i> supports generating <i>DeviceFailureEvent</i> indicating changes to individual devices in an underlying system. |
| Server | Base Info GetMonitoredItems Method | The <i>Server</i> supports obtaining subscription information via <i>GetMonitoredItems Method</i> on the <i>Server</i> object. |
| Server | Base Info Type System | The <i>Server</i> exposes a Type System with <i>DataTypes</i> , <i>ReferenceTypes</i> , <i>ObjectTypes</i> and <i>VariableTypes</i> including all of the OPC UA (namespace 0) types that are used by the <i>Server</i> , as defined in Part 5. Items that are defined in Namespace 0 but are defined in other specification parts are tested as part of the other information models. |
| Server | Base Info Custom Type System | The <i>Server</i> supports custom types (i.e. types that are derived from well-known <i>ObjectTypes</i> , <i>VariableTypes</i> , <i>ReferenceTypes</i> or <i>DataTypes</i>). Supporting this conformance unit requires that the custom types with their full inheritance tree are exposed in the <i>AddressSpace</i> . |
| Server | Base Info Model Change | The <i>Server</i> supports <i>ModelChange Event</i> and <i>NodeVersion Property</i> for all <i>Nodes</i> that the server allows Model changes for. |

| Category | Title | Description |
|----------|--|--|
| Server | Base Info Placeholder Modelling Rules | The <i>Server</i> supports defining custom <i>Object</i> or <i>Variables</i> that include the use of <i>OptionalPlaceholder</i> or <i>MandatoryPlaceholder</i> modelling rules. |
| Server | Base Info SemanticChange | The <i>Server</i> supports <i>SemanticChangeEvent</i> for some <i>Properties</i> . This includes setting the <i>SemanticChange</i> Bit in the status when a semantic change occurs, such as a change in the engineering unit associated with a value. |
| Server | Base Info EventQueueOverflowEventType | The <i>Server</i> supports the <i>EventQueueOverflowEventType</i> as defined in Part 4. |
| Server | Base Info OptionSet | The <i>Server</i> supports the <i>VariableType</i> <i>OptionSet</i> . |
| Server | Base Info ValueAsText | The <i>Server</i> supports the <i>Property</i> <i>ValueAsText</i> for enumerated <i>DataTypes</i> . |
| Server | Base Info Engineering Units | The <i>Server</i> supports defining <i>Variables</i> that include the <i>Engineering Units</i> <i>Property</i> . This property makes use of the <i>EUInformation</i> data structure. This structure by default represents the UN/CEFACT "Codes for Units of Measurement". If a different EU representation is required then the <i>EUInformation.namespaceUri</i> will indicate the alternate namespace. |
| Server | Base Info FileType Base | The <i>Server</i> supports the <i>FileType</i> <i>Object</i> (see Part 5). File writing may be restricted. |
| Server | Base Info FileType Write | The <i>Server</i> supports the <i>FileType</i> <i>Object</i> , including writing of files. Also included is the support of user access control on <i>FileType</i> <i>Object</i> . |
| Server | Base Info RequestServerStateChange Method | The <i>Server</i> supports the <i>RequestServerStateChange</i> Method. |
| Client | Base Info Client Basic | The <i>Client</i> uses the defined OPC UA <i>AddressSpace</i> . Access or provide access to <i>Server</i> information like the <i>Server's</i> state, <i>BuildInfo</i> , capabilities, <i>Namespace Table</i> and <i>Type Model</i> . |
| Client | Base Info Client Honour Operation Limits | The <i>Client</i> shall honour <i>Server</i> limits described in <i>ServerCapabilities</i> <i>Object</i> of <i>Server</i> . |
| Client | Base Info Client System Status | The <i>Client</i> makes use of <i>SystemStatusChangeEvent</i> to detect server shutdowns. |
| Client | Base Info Client System Status Underlying System | The <i>Client</i> makes use of <i>SystemStatusChangeEvent</i> to detect changes to an Underlying System (<i>SourceNode</i> = <i>Server</i>). |
| Client | Base Info Client Device Failure | The <i>Client</i> makes use of <i>DeviceFailureEventType</i> to detect failed devices in underlying systems |
| Client | Base Info Client Progress Events | The <i>Client</i> makes use of <i>ProgressEvents</i> , including checking for their support. |
| Client | Base Info Client Diagnostics | The <i>Client</i> provides interactive or programmatic access to the <i>Server's</i> diagnostic information. |
| Client | Base Info Client Type Programming | The <i>Client</i> programmatically process instances of <i>Objects</i> or <i>Variables</i> by using their type definitions. This includes custom <i>DataTypes</i> , <i>ObjectTypes</i> and <i>VariableTypes</i> . |
| Client | Base Info Client Type Pre-Knowledge | The <i>Client</i> shall interoperate with <i>Servers</i> that do not expose OPC UA Types in <i>AddressSpace</i> . |
| Client | Base Info Client Change Events | The <i>Client</i> processes <i>ModelChangeEvents</i> to detect changes in the <i>Server's</i> OPC UA <i>AddressSpace</i> and take appropriate action for a given change. |

| Category | Title | Description |
|----------|---|---|
| Client | Base Info Event Processing | The Client is able to subscribe for and process base OPC UA Events. |
| Client | Base Info Client GetMonitoredItems Method | The <i>Client</i> makes use of <i>GetMonitoredItems Method</i> to recover for communication interruptions and/or to recover subscription information. |
| Client | Base Info Client FileType Base | The <i>Client</i> can access a <i>FileType Object</i> to transfer a file from the <i>Server</i> to the <i>Client</i> . This includes large files. |
| Client | Base Info Client FileType Write | The <i>Client</i> can access a <i>FileType Object</i> to transfer a file from the <i>Client</i> to the <i>Server</i> . This includes large files. |
| Client | Base Info Client RequestServerStateChange | The Client can invoke the <i>RequestServerStateChange Method</i> . |

Table 14 describes Address Space Model information related items that can be profiled. The details of these model items are defined in Part 3 and Part 5. This includes *Server Facets* that describe what a *Server* exposes and *Client Facets* that describe what a *Client* consumes

Table 14 – Address Space model

| Category | Title | Description |
|----------|--|---|
| Server | Address Space Base | Support the <i>NodeClasses</i> with their <i>Attributes</i> and <i>References</i> as defined in Part 3. This includes for instance: <i>Object</i> , <i>ObjectType</i> , <i>Variable</i> , <i>VariableType</i> , <i>References</i> and <i>DataType</i> . |
| Server | Address Space Events | Support OPC UA <i>AddressSpace</i> elements for generating <i>Event</i> notifications. This includes at least one <i>Node</i> with an <i>EventNotifier Attribute</i> set to True (<i>Server Node</i>). |
| Server | Address Space Complex DataTypes | Support <i>StructuredDataTypes</i> with a Data Dictionary. |
| Server | Address Space Method | Support <i>Method Nodes</i> . |
| Server | Address Space Notifier Hierarchy | Supports using the <i>HasNotifier</i> reference to build a hierarchy of <i>Object Nodes</i> that are notifiers with other notifier <i>Object Nodes</i> . |
| Server | Address Space Source Hierarchy | Supports hierarchies of event sources where each hierarchy roots in an <i>Object Node</i> that is a notifier. The <i>HasEventSource Reference</i> is used to relate the <i>Nodes</i> within a hierarchy. If <i>Conditions</i> are supported, the hierarchy shall include <i>HasCondition References</i> . |
| Server | Address Space WriteMask | Supports <i>WriteMask</i> indicating the write access availability for all attributes, including not supported attributes. |
| Server | Address Space UserWriteMask | Supports <i>UserWriteMask</i> indicating the write access availability for all attributes for the given user, including not supported attributes. Support includes at least two levels of users. |
| Server | Address Space UserWriteMask Multilevel | Supports <i>UserWriteMask</i> indicating the write access availability for all attributes for the given user, including not supported attributes. This includes supporting multiple levels of access control for all nodes in the system. |
| Server | Address Space User Access Level Full | Implements User Access Level security, this includes supporting multiple levels of access control for <i>Variable</i> nodes in the system. This includes an indication of read, write, Historical read and Historical write access to the <i>Value Attribute</i> . |
| Server | Address Space User Access Level Base | Implements User Access Level Security for <i>Variable</i> nodes, this includes at least two users in the system. This includes an indication of read, write, historical read and Historical write access to the value attribute |

| Category | Title | Description |
|----------|---|--|
| Client | Address Space Client Base | Uses and understands the <i>NodeClasses</i> with their <i>Attributes</i> and behaviour as defined in Part 3. This includes for instance: <i>Object</i> , <i>ObjectType</i> , <i>Variable</i> , <i>VariableType</i> , <i>References</i> and <i>DataTypes</i> . This includes treating <i>BrowseNames</i> and <i>String NodeIds</i> as case sensitive. |
| Client | Address Space Client Complex DataTypes | Uses and understands arbitrary <i>StructuredDataTypes</i> via <i>Data Dictionary</i> . |
| Client | Address Space Client Notifier Hierarchy | Uses hierarchy of <i>Object Nodes</i> that are notifiers to detect specific areas where the <i>Client</i> can subscribe for Events. |
| Client | Address Space Client Source Hierarchy | Detect and use the hierarchy of event sources exposed for specific <i>Object Nodes</i> that are event notifiers. |

Table 15 describes Data Access information model related items that can be profiled. The details of this model are defined in Part 8. *Server* could expose this information model and *Client* could utilize this information model.

Table 15 – Data Access

| Category | Title | Description |
|----------|-----------------------------------|---|
| Server | Data Access DataItems | Provide <i>Variables</i> of <i>DataItemType</i> or one of its subtypes. Support the <i>StatusCodes</i> specified in the Part 8. Support of optional <i>Properties</i> (e.g. "InstrumentRange") shall be verified during certification testing and will be shown in the <i>Certificate</i> . |
| Server | Data Access AnalogItems | Support <i>AnalogItemType Variables</i> with corresponding <i>Properties</i> . The support of optional properties will be listed. |
| Server | Data Access PercentDeadband | Support <i>PercentDeadband</i> filter when monitoring <i>AnalogItemType Variables</i> . |
| Server | Data Access Semantic Changes | Support semantic changes of <i>AnalogItemType</i> items (<i>EURange Property</i> and/or <i>EngineeringUnits Property</i>). Support semantic change <i>StatusCode</i> bits where appropriate. |
| Server | Data Access TwoState | Support <i>TwoStateDiscreteType Variables</i> with corresponding <i>Properties</i> . |
| Server | Data Access MultiState | Support <i>MultiStateDiscreteType Variables</i> with corresponding <i>Properties</i> . |
| Server | Data Access ArrayItemType | Provide <i>Variables</i> of <i>ArrayItemType</i> or one of its subtypes (<i>YArrayItemType</i> , <i>XYArrayItemType</i> , <i>ImageArrayType</i> , <i>CubeArrayType</i> and <i>NDimensionArrayType</i>). The supported subtypes will be listed. Support for this type includes supporting all of the mandatory properties including <i>AxisInformation</i> . |
| Server | Data Access Complex Number | Supports the <i>Complex Number</i> data type. This data type is available for any variable types that do not have other explicit restrictions. |
| Server | Data Access DoubleComplex Number | Supports the <i>DoubleComplex Number</i> data type. This data type is available for any variable types that do not have other explicit restrictions. |
| Client | Data Access Client Basic | Understand the <i>DataAccess Variable Types</i> . Make use of the standard <i>Properties</i> if applicable. |
| Client | Data Access Client Deadband | Use <i>PercentDeadband</i> to filter value changes of <i>AnalogItemType Variables</i> . |
| Client | Data Access Client SemanticChange | Recognize the semantic change bit in the <i>StatusCode</i> while monitoring items and take proper action. Typically, the <i>Client</i> has to re-read <i>Properties</i> that define type-specific |

| Category | Title | Description |
|----------|-------|--|
| | | semantic like the EURange and EngineeringUnits Properties. |

Table 16 describes *Alarm* and *Conditions* information model related items that can be profiled. The details of this model are defined in Part 9. *Servers* that deal with *Alarm* and *Conditions* would expose this information model and *Clients* that process *Alarms* and *Conditions* would utilize this information model.

Table 16 – Alarms and Conditions

| Category | Title | Description |
|----------|----------------------------------|---|
| Server | A & C Basic | Supports <i>Alarm & Condition</i> model <i>ConditionType</i> . |
| Server | A & C Enable | Supports Enable and Disable Methods. |
| Server | A & C Refresh | Supports <i>ConditionRefresh Method</i> and the concept of a refresh. |
| Server | A & C Refresh2 | Supports <i>ConditionRefresh2 Method</i> and the concept of a monitored item based refresh. |
| Server | A & C Instances | Support exposing of A&C <i>Condition</i> instances in the <i>AddressSpace</i> . |
| Server | A & C ConditionClasses | Supports multiple <i>Condition</i> classes for grouping and filtering of <i>Alarms</i> . |
| Server | A & C Acknowledge | Supports the Acknowledge concept, <i>Acknowledge Method</i> , and <i>AcknowledgeableCondition Type</i> . |
| Server | A & C Confirm | Supports the concept of Confirm and the <i>Confirm Method</i> . |
| Server | A & C Comment | Supports the concept of Comments and the <i>AddComment Method</i> . |
| Server | A & C Alarm | Supports the mandatory features of the <i>AlarmCondition Type</i> . |
| Server | A & C Branch | Support for branching of Condition Types and any subtypes, such as <i>AcknowledgeableConditionType</i> and <i>AlarmConditionType</i> etc. |
| Server | A & C Shelving | Support the concept of shelving and the <i>TimedShelve</i> , <i>OneShotShelve</i> and <i>Unshelve Methods</i> . |
| Server | A & C Exclusive Level | Supports Exclusive Level <i>Alarm</i> type. |
| Server | A & C Exclusive Limit | Supports Exclusive Limit <i>Alarms</i> . A <i>Server</i> that supports this must support at least one of the sub-types: Level, Deviation or RateofChange. |
| Server | A & C Exclusive Deviation | Supports Exclusive Deviation <i>Alarm</i> type. |
| Server | A & C Exclusive RateofChange | Supports Exclusive RateofChange <i>Alarm</i> type. |
| Server | A & C Non-Exclusive Limit | Supports Non-Exclusive Limit <i>Alarms</i> . A <i>Server</i> that supports this must support at least one of the sub-types: Level, Deviation or RateofChange. |
| Server | A & C Non-Exclusive Level | Supports Non-Exclusive Level <i>Alarm</i> type. |
| Server | A & C Non-Exclusive Deviation | Supports Non-Exclusive Deviation <i>Alarm</i> type. |
| Server | A & C Non-Exclusive RateofChange | Supports Non-Exclusive RateofChange <i>Alarm</i> type. |
| Server | A & C Discrete | Supports Discrete <i>Alarm</i> types. |
| Server | A & C OffNormal | Supports <i>OffNormalAlarmType</i> . |

| Category | Title | Description |
|----------|---|---|
| Server | A & C SystemOffNormal | Supports <i>SystemOffNormalAlarmType</i> . |
| Server | A & C Trip | Supports Trip <i>Alarm</i> type. |
| Server | A & C Dialog | Supports DialogConditionType including Respond <i>Method</i> . |
| Server | A & C CertificateExpiration | Supports CertificateExpirationAlarmType. |
| Server | A & E Wrapper Mapping | The Server uses the COM A&E mapping specified in the annex of Part 9 to map OPC-COM Events to A&C Events. This includes <i>Condition</i> Class mapping. |
| Client | A & C Basic Client | Uses the <i>Alarm & Condition</i> model ConditionType. |
| Client | A & C Enable Client | Uses Enable and Disable Methods. |
| Client | A & C Refresh Client | Uses ConditionRefresh <i>Method</i> and the concept of a refresh. |
| Client | A & C Refresh2 Client | Uses ConditionRefresh2 <i>Method</i> and the concept of a monitored item based refresh. |
| Client | A & C Instances Client | Uses A&C <i>Condition</i> instances when they are exposed in the <i>AddressSpace</i> . |
| Client | A & C ConditionClasses Client | Uses <i>Condition</i> classes to group <i>Alarms</i> . |
| Client | A & C Acknowledge Client | Understands the Acknowledge concept and the <i>AcknowledgeableCondition Type</i> , and uses the <i>Acknowledge Method</i> if requested. |
| Client | A & C Confirm Client | Understands the concept of confirming <i>Conditions</i> and uses the Confirm <i>Method</i> . |
| Client | A & C Comment Client | Understands the concept of Comments and uses the AddComment <i>Method</i> . |
| Client | A & C Alarm Client | Understands the concept of <i>Alarms</i> and uses the mandatory features of the <i>AlarmCondition Type</i> , |
| Client | A & C Branch Client | Can make use of and process <i>Condition</i> Branches, including all actions associated with previous <i>Condition</i> instances. |
| Client | A & C Shelving Client | Understand the shelving model and use the TimedShelve, OneShotShelve and Unshelve <i>Methods</i> . |
| Client | A & C Exclusive Level Client | Uses Exclusive Level <i>Alarms</i> . |
| Client | A & C Exclusive Limit Client | Uses Exclusive Limit <i>Alarms</i> . Requires that at least one of the sub-types be used. |
| Client | A & C Exclusive Deviation Client | Uses Exclusive Deviation <i>Alarms</i> . |
| Client | A & C Exclusive RateofChange Client | Uses Exclusive RateofChange <i>Alarms</i> . |
| Client | A & C Non-Exclusive Level Client | Uses Non-Exclusive Level <i>Alarms</i> . |
| Client | A & C Non-Exclusive Limit Client | Uses Non-Exclusive Limit <i>Alarms</i> . Requires that at least one of the sub-types be used. |
| Client | A & C Non-Exclusive Deviation Client | Uses Non-Exclusive Deviation <i>Alarms</i> . |
| Client | A & C Non-Exclusive RateofChange Client | Uses Non-Exclusive RateofChange <i>Alarms</i> . |

| Category | Title | Description |
|----------|------------------------------------|---|
| Client | A & C Discrete Client | Uses Discrete <i>Alarm</i> types. |
| Client | A & C OffNormal Client | Uses <i>OffNormalAlarmType</i> . |
| Client | A & C SystemOffNormal Client | Uses <i>SystemOffNormalAlarmType</i> . |
| Client | A & C Trip Client | Uses <i>TripAlarmType</i> . |
| Client | A & C Dialog Client | Uses <i>DialogConditionType</i> including Respond <i>Method</i> . |
| Client | A & C CertificateExpiration Client | Uses <i>CertificateExpirationAlarmType</i> . |

Table 17 describes Historical Data Access information model related items that can be profiled. The details of this model are defined in Part 11. *Servers* that support some level of historical data would expose this information model and *Clients* that utilize historical data would utilize this information model.

Table 17 – Historical Access

| Category | Title | Description |
|----------|--|---|
| Server | Historical Access Read Raw | General support for basic historical access, reading raw data using the ReadRawModifiedDetails structure. Where the time range is specified using a start time, stop time and number of values (a minimum of two of the three parameters must be provided) and the ReadModified flag is set to False. |
| Server | Historical Access Data Max Nodes Read Continuation Point | Supports enough continuation points to cover the number of supported points indicated in the MaxNodesPerHistoryReadData Server OperationLimits parameter for historical data access. |
| Server | Historical Access Time Instance | Supports reading historical data at a specified instance in time using the ReadAtTimeDetails structure. |
| Server | Historical Access Aggregates | Supports reading one or more Aggregates of historical values of <i>Variables</i> using the ReadProcessedDetails structure. At least one of the Aggregates described in Part 13 must be supported. |
| Server | Historical Access Insert Value | Supports inserting historical values of <i>Variables</i> . |
| Server | Historical Access Delete Value | Supports deleting historical values of <i>Variables</i> . |
| Server | Historical Access Update Value | Supports updating historical values of <i>Variables</i> . |
| Server | Historical Access Replace Value | Supports replacing historical values of <i>Variables</i> . |
| Server | Historical Access Modified Values | Supports maintaining old values for historical data that have been updated and the retrieval of these values using the ReadRawModifiedDetails structure (ReadModified flag set to true). |
| Server | Historical Access Annotations | Supports the entry and retrieval of Annotations for historical data. The retrieval is accomplished using the standard historical read raw functionality (ReadRawModifiedDetails). The entry uses the standard historical update (UpdateStructureDataDetails) functionality. |
| Server | Historical Access ServerTimestamp | Supports providing a ServerTimestamp (as well as the default SourceTimestamp). |

| Category | Title | Description |
|----------|--|---|
| Server | Historical Access Structured Data Read Raw | Supports ReadRawModified historical access for structured data. Supporting the structure for an annotation is not considered supporting generic structured data. |
| Server | Historical Access Structured Data Time Instance | Supports historical access for structured data. Supporting ReadAtTimeDetails for structured data. Supporting the structure for an annotation is not considered supporting generic structured data. |
| Server | Historical Access Structured Data Insert | Supports historical access for structured data. Inserting Structured data. Supporting the structure for an annotation is not considered supporting generic structured data. |
| Server | Historical Access Structured Data Delete | Supports historical access for structured data. Delete of existing data. Supporting the structure for an annotation is not considered supporting generic structured data. |
| Server | Historical Access Structured Data Update | Supports historical access for structured data. Updates of existing data. Supporting the structure for an annotation is not considered supporting generic structured data. |
| Server | Historical Access Structured Data Replace | Supports replacing structured historical data. Supporting the structure for an annotation is not considered supporting generic structured data. |
| Server | Historical Access Structured Data Read Modified | Supports maintaining old values for historical structured data that have been updated and the retrieval of these values. Using the ReadRawModifiedDetails structure (ReadModified flag set to true) for structured data. Supporting the structure for an annotation is not considered supporting generic structured data. |
| Server | Historical Access Events | Supports the retrieval of historical Events using the ReadEventDetails structure. This includes support for simple filtering of Events. The <i>Event</i> fields that are stored are server specific, but at least the mandatory fields of BaseEventType are required. |
| Server | Historical Access Event Max Events Read Continuation Point | Supports enough continuation points to cover the number of supported <i>Event</i> reads indicated in the MaxNodesPerHistoryReadEvents Server OperationLimits parameter for Historical <i>Event</i> access. |
| Server | Historical Access Insert Event | Supports inserting historical Events. |
| Server | Historical Access Update Event | Supports updating historical Events. |
| Server | Historical Access Replace Event | Supports replacing historical Events. |
| Server | Historical Access Delete Event | Supports deleting of historical Events. |
| Client | Historical Access Client Browse | Uses the View <i>Service</i> Set to discover <i>Nodes</i> with historical data. |
| Client | Historical Access Client Read Raw | Uses the HistoryRead <i>Service</i> to read raw historical data using the ReadRawModifiedDetails Structure (ReadModified Flag set to False). |
| Client | Historical Access Client Read Modified | Uses the HistoryRead <i>Service</i> to read modified historical data using the ReadRawModifiedDetails Structure (ReadModified Flag set to True). |
| Client | Historical Access Client Read Aggregates | Uses the HistoryRead <i>Service</i> to read Aggregated historical data. This includes using at least one of the Aggregates defined in Part 13. |
| Client | Historical Access Client Structure Data Raw | Uses the HistoryRead <i>Service</i> to read raw historical data using the ReadRawModifiedDetails Structure (ReadModified Flag set to False) for structured data. |
| Client | Historical Access Client Structure | Uses the HistoryRead <i>Service</i> to read modified structured historical data using the ReadRawModifiedDetails Structure (ReadModified Flag set to True). |

| Category | Title | Description |
|----------|---|---|
| | Data Read Modified | |
| Client | Historical Access Client Structure Data Insert | Uses the HistoryUpdate <i>Service</i> to insert historical data values for structured data. |
| Client | Historical Access Client Structure Data Delete | Uses the HistoryUpdate <i>Service</i> to delete historical data values for structured data. |
| Client | Historical Access Client Structure Data Update | Uses the HistoryUpdate <i>Service</i> to update historical data values for structured data. |
| Client | Historical Access Client Structure Data Replace | Uses the HistoryUpdate <i>Service</i> to replace historical data values for structured data. |
| Client | Historical Access Client Structure Data Time Instance | Reads historical data at a specified instance in time for structured data. Using the ReadAtTimeDetails structure. |
| Client | Historical Access Client Read Events | Uses the HistoryRead <i>Service</i> to read historical <i>Event</i> data using the ReadEventDetails Structure. |
| Client | Historical Access Client Event Inserts | Uses the HistoryUpdate <i>Service</i> to insert historical Events. |
| Client | Historical Access Client Event Updates | Uses the HistoryUpdate <i>Service</i> to update historical Events. |
| Client | Historical Access Client Event Replaces | Uses the HistoryUpdate <i>Service</i> to replace historical Events. |
| Client | Historical Access Client Event Deletes | Uses the HistoryUpdate <i>Service</i> to delete historical Events. |
| Client | Historical Access Client Data Insert | Uses the HistoryUpdate <i>Service</i> to insert historical data values. |
| Client | Historical Access Client Data Delete | Uses the HistoryUpdate <i>Service</i> to delete historical data values. |
| Client | Historical Access Client Data Update | Uses the HistoryUpdate <i>Service</i> to update historical data values. |
| Client | Historical Access Client Data Replace | Uses the HistoryUpdate <i>Service</i> to replace historical data values. |
| Client | Historical Access Client Annotations | Enters and retrieves Annotations of historical data. The retrieval is accomplished using the standard historical read raw functionality (ReadRawModifiedDetails). The entry uses the standard Historical Update (UpdateStructureDataDetails) functionality. |
| Client | Historical Access Client Time Instance | Reads historical data at a specified instance in time using the ReadAtTimeDetails structure. |
| Client | Historical Access Client Server Timestamp | Uses the ServerTimestamp (as well as the default SourceTimestamp), if it is provided by the <i>Server</i> . |

Table 18 describes Aggregate related items that can be profiled. *Servers* that support the Aggregates would expose this functionality and *Clients* that utilize Aggregates would implement some of this functionality.

Table 18 – Aggregates

| Category | Title | Description |
|----------|------------------------------------|--|
| Server | Aggregate Master Configuration | Supports an AggregateConfigurationType <i>Object</i> as part of the HistoricalServerCapabilities (defined in UA Part 11). |
| Server | Aggregate Historical Configuration | Supports at least one AggregateConfigurationType <i>Object</i> . AggregateConfigurationType <i>Objects</i> occur as part of an HistoricalConfiguration Object, allowing <i>Variable</i> specific configurations. |
| Server | Aggregate Interpolative – | Supports the Interpolative Aggregate for Historical access. |
| Server | Aggregate Average – | Supports the Average Aggregate for Historical access. |
| Server | Aggregate TimeAverage – | Supports the TimeAverage Aggregate for Historical access. |
| Server | Aggregate TimeAverage2 – | Supports the TimeAverage2 Aggregate for Historical access. |
| Server | Aggregate – Total | Supports the Total Aggregate for Historical access. |
| Server | Aggregate Total2 – | Supports the Total2 Aggregate for Historical access. |
| Server | Aggregate Minimum – | Supports the Minimum Aggregate for Historical access. |
| Server | Aggregate MinimumActualTime – | Supports the MinimumActualTime Aggregate for Historical access. |
| Server | Aggregate Minimum2 – | Supports the Minimum2 Aggregate for Historical access. |
| Server | Aggregate MinimumActualTime2 – | Supports the MinimumActualTime2 Aggregate for Historical access. |
| Server | Aggregate Maximum – | Supports the Maximum Aggregate for Historical access. |
| Server | Aggregate MaximumActualTime – | Supports the MaximumActualTime Aggregate for Historical access. |
| Server | Aggregate Maximum2 – | Supports the Maximum2 Aggregate for Historical access. |
| Server | Aggregate MaximumActualTime2 – | Supports the MaximumActualTime2 Aggregate for Historical access. |
| Server | Aggregate Range – | Supports the Range Aggregate for Historical access. |
| Server | Aggregate Range2 – | Supports the Range2 Aggregate for Historical access. |
| Server | Aggregate Count – | Supports the Count Aggregate for Historical access. |
| Server | Aggregate DurationInStateZero – | Supports the DurationInStateZero Aggregate for Historical access. |
| Server | Aggregate DurationInStateNonZero – | Supports the DurationInStateNonZero Aggregate for Historical access. |
| Server | Aggregate NumberOfTransitions – | Supports the NumberOfTransitions Aggregate for Historical access. |
| Server | Aggregate – Start | Supports the Start Aggregate for Historical access. |
| Server | Aggregate StartBound – | Supports the StartBound Aggregate for Historical access. |
| Server | Aggregate – End | Supports the End Aggregate for Historical access. |

| Category | Title | Description |
|----------|---|--|
| Server | Aggregate EndBound – | Supports the EndBound Aggregate for Historical access. |
| Server | Aggregate – Delta | Supports the Delta Aggregate for Historical access. |
| Server | Aggregate DeltaBounds – | Supports the DeltaBounds Aggregate for Historical access. |
| Server | Aggregate DurationGood – | Supports the DurationGood Aggregate for Historical access. |
| Server | Aggregate DurationBad – | Supports the DurationBad Aggregate for Historical access. |
| Server | Aggregate PercentGood – | Supports the PercentGood Aggregate for Historical access. |
| Server | Aggregate PercentBad – | Supports the PercentBad Aggregate for Historical access. |
| Server | Aggregate WorstQuality – | Supports the WorstQuality Aggregate for Historical access. |
| Server | Aggregate WorstQuality2 – | Supports the WorstQuality2 Aggregate for Historical access. |
| Server | Aggregate AnnotationCount – | Supports the AnnotationCount Aggregate for Historical access. |
| Server | Aggregate StandardDeviationSample – | Supports the StandardDeviationSample Aggregate for Historical access. |
| Server | Aggregate VarianceSample – | Supports the VarianceSample Aggregate for Historical access. |
| Server | Aggregate StandardDeviationPopulation – | Supports the StandardDeviationPopulation for Historical access. |
| Server | Aggregate VariancePopulation – | Supports the VariancePopulation for Historical access. |
| Server | Aggregate Custom – | The <i>Server</i> supports custom Aggregates for Historical access that do not have standard tests defined. These Aggregates are list as untested by this <i>ConformanceUnit</i> . |
| Server | Aggregate Subscription Filter – | Supports Aggregate subscription filters which requires at least one of the defined Aggregates is supported as defined in Part 13. |
| Server | Aggregate Subscription Interpolative – | Supports subscription filter for the Interpolative Aggregate. |
| Server | Aggregate Subscription Average – | Supports subscription filter for the Average Aggregate. |
| Server | Aggregate Subscription TimeAverage – | Supports subscription filter for the TimeAverage Aggregate. |
| Server | Aggregate Subscription TimeAverage2 – | Supports subscription filter for the TimeAverage2 Aggregate. |
| Server | Aggregate Subscription Total – | Supports subscription filter for the Total Aggregate. |
| Server | Aggregate Subscription Total2 – | Supports subscription filter for the Total2 Aggregate. |
| Server | Aggregate Subscription Minimum – | Supports subscription filter for the Minimum Aggregate. |
| Server | Aggregate Subscription – | Supports subscription filter for the MinimumActualTime Aggregate. |

| Category | Title | Description |
|----------|---|--|
| | MinimumActualTime | |
| Server | Aggregate Subscription – Minimum2 | Supports subscription filter for the Minimum2 Aggregate. |
| Server | Aggregate Subscription – MinimumActualTime2 | Supports subscription filter for the MinimumActualTime2 Aggregate. |
| Server | Aggregate Subscription – Maximum | Supports subscription filter for the Maximum Aggregate. |
| Server | Aggregate Subscription – MaximumActualTime | Supports subscription filter for the MaximumActualTime Aggregate. |
| Server | Aggregate Subscription – Maximum2 | Supports subscription filter for the Maximum2 Aggregate. |
| Server | Aggregate Subscription – MaximumActualTime2 | Supports subscription filter for the MaximumActualTime2 Aggregate. |
| Server | Aggregate Subscription – Range | Supports subscription filter for the Range Aggregate. |
| Server | Aggregate Subscription – Range2 | Supports subscription filter for the Range2 Aggregate. |
| Server | Aggregate Subscription – Count | Supports subscription filter for the Count Aggregate. |
| Server | Aggregate Subscription – DurationInStateZero | Supports subscription filter for the DurationInStateZero Aggregate. |
| Server | Aggregate Subscription – DurationInStateNonZero | Supports subscription filter for the DurationInStateNonZero Aggregate. |
| Server | Aggregate Subscription – NumberOfTransitions | Supports subscription filter for the NumberOfTransitions Aggregate. |
| Server | Aggregate Subscription – Start | Supports subscription filter for the Start Aggregate. |
| Server | Aggregate Subscription – StartBound | Supports subscription filter for the StartBound Aggregate. |
| Server | Aggregate Subscription – End | Supports subscription filter for the End Aggregate. |
| Server | Aggregate Subscription – EndBound | Supports subscription filter for the EndBound Aggregate. |
| Server | Aggregate Subscription – Delta | Supports subscription filter for the Delta Aggregate. |

| Category | Title | Description |
|----------|--|---|
| Server | Aggregate Subscription DeltaBounds – | Supports subscription filter for the DeltaBounds Aggregate. |
| Server | Aggregate Subscription DurationGood – | Supports subscription filter for the DurationGood Aggregate. |
| Server | Aggregate Subscription DurationBad – | Supports subscription filter for the DurationBad Aggregate. |
| Server | Aggregate Subscription PercentGood – | Supports subscription filter for the PercentGood Aggregate. |
| Server | Aggregate Subscription PercentBad – | Supports subscription filter for the PercentBad Aggregate. |
| Server | Aggregate Subscription WorstQuality – | Supports subscription filter for the WorstQuality Aggregate. |
| Server | Aggregate Subscription WorstQuality2 – | Supports subscription filter for the WorstQuality2 Aggregate. |
| Server | Aggregate Subscription AnnotationCount – | Supports subscription filter for the AnnotationCount Aggregate. |
| Server | Aggregate Subscription StandardDeviationSample – | Supports subscription filter for the StandardDeviationSample Aggregate. |
| Server | Aggregate Subscription VarianceSample – | Supports subscription filter for the VarianceSample Aggregate. |
| Server | Aggregate Subscription StandardDeviationPopulation – | Supports subscription filter for the StandardDeviationPopulation Aggregate. |
| Server | Aggregate Subscription VariancePopulation – | Supports subscription filter for the VariancePopulation Aggregate. |
| Server | Aggregate Subscription Custom – | The <i>Server</i> supports subscribing to custom Aggregates that do not have standard tests defined. These Aggregates are listed as untested by this <i>ConformanceUnit</i> . |
| Client | Aggregate Client Usage – | Uses Historical access to Aggregate which requires at least one of the defined Aggregates is supported as defined in Part 13. |
| Client | Aggregate Client Interpolative – | Uses Historical access to the Interpolative Aggregate. |
| Client | Aggregate Client Average – | Uses Historical access to the Average Aggregate. |
| Client | Aggregate Client TimeAverage – | Uses Historical access to the TimeAverage Aggregate. |
| Client | Aggregate Client TimeAverage2 – | Uses Historical access to the TimeAverage2 Aggregate. |
| Client | Aggregate Client Total – | Uses Historical access to the Total Aggregate. |

| Category | Title | Description |
|----------|---|---|
| Client | Aggregate – Client Total2 | Uses Historical access to the Total2 Aggregate. |
| Client | Aggregate – Client Minimum | Uses Historical access to the Minimum Aggregate. |
| Client | Aggregate – Client MinimumActualTi me | Uses Historical access to the MinimumActualTime Aggregate. |
| Client | Aggregate – Client Minimum2 | Uses Historical access to the Minimum2 Aggregate. |
| Client | Aggregate – Client MinimumActualTi me2 | Uses Historical access to the MinimumActualTime2 Aggregate. |
| Client | Aggregate – Client Maximum | Uses Historical access to the Maximum Aggregate. |
| Client | Aggregate – Client MaximumActualTi me | Uses Historical access to the MaximumActualTime Aggregate. |
| Client | Aggregate – Client Maximum2 | Uses Historical access to the Maximum2 Aggregate. |
| Client | Aggregate – Client MaximumActualTi me2 | Uses Historical access to the MaximumActualTime2 Aggregate. |
| Client | Aggregate – Client Range | Uses Historical access to the Range Aggregate. |
| Client | Aggregate – Client Range2 | Uses Historical access to the Range2 Aggregate. |
| Client | Aggregate – Client Count | Uses Historical access to the Count Aggregate. |
| Client | Aggregate – Client DurationInStateZ ero | Uses Historical access to the DurationInStateZero Aggregate. |
| Client | Aggregate – Client DurationInStateN onZero | Uses Historical access to the DurationInStateNonZero Aggregate. |
| Client | Aggregate – Client NumberOfTransiti ons | Uses Historical access to the NumberOfTransitions Aggregate. |
| Client | Aggregate – Client Start | Uses Historical access to the Start Aggregate. |
| Client | Aggregate – Client StartBound | Uses Historical access to the StartBound Aggregate. |
| Client | Aggregate – Client End | Uses Historical access to the End Aggregate. |
| Client | Aggregate – Client EndBound | Uses Historical access to the EndBound Aggregate. |
| Client | Aggregate – Client Delta | Uses Historical access to the Delta Aggregate. |
| Client | Aggregate – Client DeltaBounds | Uses Historical access to the DeltaBounds Aggregate. |

| Category | Title | Description |
|----------|--|---|
| Client | Aggregate – Client DurationGood | Uses Historical access to the DurationGood Aggregate. |
| Client | Aggregate – Client DurationBad | Uses Historical access to the DurationBad Aggregate. |
| Client | Aggregate – Client PercentGood | Uses Historical access to the PercentGood Aggregate. |
| Client | Aggregate – Client PercentBad | Uses Historical access to the PercentBad Aggregate. |
| Client | Aggregate – Client WorstQuality | Uses Historical access to the WorstQuality Aggregate. |
| Client | Aggregate – Client WorstQuality2 | Uses Historical access to the WorstQuality2 Aggregate. |
| Client | Aggregate – Client AnnotationCount | Uses Historical access to the AnnotationCount Aggregate. |
| Client | Aggregate – Client StandardDeviationSample | Uses Historical access to the StandardDeviationSample Aggregate. |
| Client | Aggregate – Client VarianceSample | Uses Historical access to the VarianceSample Aggregate. |
| Client | Aggregate – Client StandardDeviationPopulation | Uses Historical access to the StandardDeviationPopulation Aggregate. |
| Client | Aggregate – Client VariancePopulation | Uses Historical access to the VariancePopulation Aggregate. |
| Client | Aggregate – Client Custom Aggregates | The <i>Client</i> can make use of all custom Aggregates in the list of Aggregates, via Historical access, exposed by the <i>Server</i> . This includes displaying or utilizing the data in some manner. |
| Client | Aggregate – Subscription Client Filter | Subscribes for data using Aggregate filters which requires at least one of the Aggregates defined in Part 13 is supported. |
| Client | Aggregate – Subscription Client Interpolative | Subscribes for data using the Interpolative Aggregate filter. |
| Client | Aggregate – Subscription Client Average | Subscribes for data using the Average Aggregate filter. |
| Client | Aggregate – Subscription Client TimeAverage | Subscribes for data using the TimeAverage Aggregate filter. |
| Client | Aggregate – Subscription Client TimeAverage2 | Subscribes for data using the TimeAverage2 Aggregate filter. |

| Category | Title | Description |
|----------|--|--|
| Client | Aggregate Subscription – Client Total | Subscribes for data using the Total Aggregate filter. |
| Client | Aggregate Subscription – Client Total2 | Subscribes for data using the Total2 Aggregate filter. |
| Client | Aggregate Subscription – Client Minimum | Subscribes for data using the Minimum Aggregate filter. |
| Client | Aggregate Subscription – Client MinimumActualTime | Subscribes for data using the MinimumActualTime Aggregate filter. |
| Client | Aggregate Subscription – Client Minimum2 | Subscribes for data using the Minimum2 Aggregate filter. |
| Client | Aggregate Subscription – Client MinimumActualTime2 | Subscribes for data using the MinimumActualTime2 Aggregate filter. |
| Client | Aggregate Subscription – Client Maximum | Subscribes for data using the Maximum Aggregate filter. |
| Client | Aggregate Subscription – Client MaximumActualTime | Subscribes for data using the MaximumActualTime Aggregate filter. |
| Client | Aggregate Subscription – Client MaximumActualTime2 | Subscribes for data using the MaximumActualTime2 Aggregate filter. |
| Client | Aggregate Subscription – Client Maximum2 | Subscribes for data using the Maximum2 Aggregate filter. |
| Client | Aggregate Subscription – Client Range | Subscribes for data using the Range Aggregate filter. |
| Client | Aggregate Subscription – Client Range2 | Subscribes for data using the Range2 Aggregate filter. |
| Client | Aggregate Subscription – Client Count | Subscribes for data using the Count Aggregate filter. |
| Client | Aggregate Subscription – Client DurationInStateZero | Subscribes for data using the DurationInStateZero Aggregate filter. |
| Client | Aggregate Subscription – Client DurationInStateNonZero | Subscribes for data using the DurationInStateNonZero Aggregate filter. |
| Client | Aggregate Subscription – | Subscribes for data using the NumberOfTransitions Aggregate filter. |

| Category | Title | Description |
|----------|--|---|
| | Client NumberOfTransitions | |
| Client | Aggregate Subscription – Client Start | Subscribes for data using the Start Aggregate filter. |
| Client | Aggregate Subscription – Client StartBound | Subscribes for data using the StartBound Aggregate filter. |
| Client | Aggregate Subscription – Client End | Subscribes for data using the End Aggregate filter. |
| Client | Aggregate Subscription – Client EndBound | Subscribes for data using the EndBound Aggregate filter. |
| Client | Aggregate Subscription – Client Delta | Subscribes for data using the Delta Aggregate filter. |
| Client | Aggregate Subscription – Client DeltaBounds | Subscribes for data using the DeltaBounds Aggregate filter. |
| Client | Aggregate Subscription – Client DurationGood | Subscribes for data using the DurationGood Aggregate filter. |
| Client | Aggregate Subscription – Client DurationBad | Subscribes for data using the DurationBad Aggregate filter. |
| Client | Aggregate Subscription – Client PercentGood | Subscribes for data using the PercentGood Aggregate filter. |
| Client | Aggregate Subscription – Client PercentBad | Subscribes for data using the PercentBad Aggregate filter. |
| Client | Aggregate Subscription – Client WorstQuality | Subscribes for data using the WorstQuality Aggregate filter. |
| Client | Aggregate Subscription – Client WorstQuality2 | Subscribes for data using the WorstQuality2 Aggregate filter. |
| Client | Aggregate Subscription – Client AnnotationCount | Subscribes for data using the AnnotationCount Aggregate filter. |
| Client | Aggregate Subscription – Client StandardDeviationSample | Subscribes for data using the StandardDeviationSample Aggregate filter. |
| Client | Aggregate Subscription – Client VarianceSample | Subscribes for data using the VarianceSample Aggregate filter. |

| Category | Title | Description |
|----------|---|---|
| Client | Aggregate Subscription – Client StandardDeviationPopulation | Subscribes for data using the StandardDeviationPopulation Aggregate filter. |
| Client | Aggregate Subscription – Client VariancePopulation | Subscribes for data using the VariancePopulation Aggregate filter. |
| Client | Aggregate Subscription – Client Custom Aggregates | The <i>Client</i> supports subscribing to all custom Aggregates in the list of Aggregates exposed by the <i>Server</i> . This includes displaying or utilizing the data in some manner. |

Table 19 describes auditing related items that can be profiled. Most full function *Servers* would support these features, although some resource constrained *Servers* may not provide this functionality. *Clients* that are security aware or are used to support security logging would support these features

Table 19 – Auditing

| Category | Title | Description |
|----------|----------------------------|---|
| Server | Auditing Base | Support AuditEvents. The list of supported AuditEvents shall be verified during certification testing and will be shown in the certification test result. Base AuditEvents are defined in Part 3 and in Part 5. |
| Client | Auditing Client Audit ID | <i>Client</i> supports generating AuditEvents ids and providing them to <i>Servers</i> . |
| Client | Auditing Client Subscribes | The <i>Client</i> supports subscribing for AuditEvents and storing / processing them in a secure manner. |

Table 20 describes Redundancy related items that are profiled. *Servers* that support redundancy would support appropriate *ConformanceUnits* based on the type of redundancy they support. *Clients* that are capable of handling redundancy would support the appropriate *ConformanceUnits* based of the type of redundancy they support.

Table 20 – Redundancy

| Category | Title | Description |
|----------|-------------------------------|---|
| Server | Redundancy Server | Supports <i>Server</i> based redundancy. |
| Server | Redundancy Server Transparent | Supports transparent <i>Server</i> redundancy. |
| Client | Redundancy Client | <i>Client</i> supports <i>Client</i> redundancy. <i>Clients</i> that support <i>Client</i> redundancy can failover to another <i>Client</i> (requires some out of band communication). |
| Client | Redundancy Client Switch | <i>Clients</i> supporting this <i>ConformanceUnit</i> monitor the redundancy status for non-transparent redundancy <i>Servers</i> and switch to the backup <i>Server</i> when they recognize a change in server status. |

Table 21 describes items for a Global Discovery Server (GDS). *Servers* that act as a GDS would support these *ConformanceUnits*.

Table 21 – Global Discovery Server

| Category | Title | Description |
|----------|------------------------------------|---|
| GDS | GDS Application Directory | Supports the Directory Object with all <i>Methods</i> like <i>RegisterApplication</i> and <i>QueryServers</i> . |
| GDS | GDS LDS-ME Connectivity | The GDS can be configured to use specific LDS-ME installations for semi-automatic application registration for all <i>Servers</i> on a subnet. |
| GDS | GDS Certificate Manager Pull Model | This Conformance Unit requires support of the complete <i>Information Model</i> and <i>Services</i> for <i>Certificate</i> management including the Pull Model as specified in Part 12. |
| GDS | GDS Certificate Manager Push Model | This Conformance Unit requires use of the complete <i>Information model</i> and <i>Services</i> for the <i>Certificate</i> management Push Model as specified in UA Part 12. |
| | | |

5.5 Miscellaneous

The following table describes miscellaneous *ConformanceUnits*.

Each table includes a listing of the *Profile Category* to which a *ConformanceUnit* belongs, the title and description of the *ConformanceUnit* and a column that indicates if the *ConformanceUnit* is derived from another *ConformanceUnit*. A *ConformanceUnit* that is derived from another *ConformanceUnit* includes all of the same tests as its parent plus one or more additional *TestCases*. These *TestCases* can only further restrict the existing *TestCases*.

Table 22 – Miscellaneous

| Category | Title | Description |
|----------------|--|---|
| Client, Server | Documentation – Supported Profiles | The documentation includes a description of the profiles supported by the product. This description includes the level of Certification testing the product has passed. |
| Client, Server | Documentation – Multiple Languages | The documentation is available in multiple languages. The results of this conformance unit include the list of supported languages. |
| Client, Server | Documentation – Users Guide | The application includes documentation that describes the available functionality provided by the application. For <i>Servers</i> it includes a summary of all functionality provided by the <i>Server</i> . |
| Client, Server | Documentation – On-line | The documentation provided by the application is available in electronic format as part of the application. The electronic documentation could be a WEB page, installed document or CD/DVD, but in all case it can be accessed from the application or from a link installed with the application. |
| Client, Server | Documentation – Installation | The application includes installation instructions that are sufficient to easily install the application. This includes descriptions of any and all possible configuration items. Instructions for loading or configuring security related items such as Application Instance Certificates. |
| Client, Server | Documentation – Trouble Shooting Guide | The application includes documentation that describes typical problems a user may encounter and actions that the user could perform to resolve the problem. It could also describe tip, tricks or other actions that could help a user diagnose or fix a problem. It could also describe tools or other items that can be used in diagnosing or repairing problems. The actual Trouble Shooting Guide can be part of other documentation, but should be complete enough to provide useful information to a novice user. |

6 Profiles

6.1 Overview

This section includes a listing of the categories that a *Profile* can be grouped into, a list of named *Profiles* and the detailed listing of each *Profile* including directly defined *ConformanceUnits* and any sub *Profiles* that are included in the *Profile*.

6.2 Profile list

Table 23 lists *Profiles*. The *Profile* table is ordered by *Profile* category and then alphabetically by the name of the *Profile*. The table includes a list of categories the *Profile* is associated with and a URI. The URI is used to uniquely identify a *Profile*. The URI shall be able to be used to access the information provided in this document with regard to the given *Profile* in an on-line display.

An application (*Client* or *Server*) shall implement all of the *ConformanceUnits* in a *Profile* in order to be compliant with the *Profile*. Some *Profiles* contain optional *ConformanceUnits*. An optional *ConformanceUnit* means that an application has the option to not support the *ConformanceUnit*. However, if supported, the application shall pass all tests associated with the *ConformanceUnit*. For example, some *ConformanceUnits* require specific information model items to be available. They are, therefore, listed as optional in order to allow for the information model items to be omitted. If a *Server* desires to be listed as supporting the optional *ConformanceUnit* then it shall include any required information model items in the configuration provided for certification testing. The test result that is generated by the certification testing lists all optional *ConformanceUnits* and whether they are supported or not by the tested UA application. Some *ConformanceUnits* also include lists of supported DataTypes or optional Subtypes, the list are handled in the same manner as optional *ConformanceUnits*. All reporting requirements for optional *ConformanceUnits* also apply to these lists of supported DataTypes or Subtypes.

Table 23 – Profile list

| Profile | Related Category | URI |
|---|------------------|---|
| Core Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/CoreFacet |
| Global Certificate Management Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/GlobalCertificateManagement |
| Subnet Discovery Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/SubnetDiscovery |
| Base Server Behaviour Facet | Server | http://opcfoundation.org/UA-Profile/Server/Behaviour |
| Request State Change Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/RequestStateChange |
| Attribute WriteMask Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/AttributeWriteMask |
| File Access Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/FileAccess |
| Documentation – Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/Documentation |
| Embedded DataChange Subscription Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/EmbeddedDataChangeSubscription |
| Standard DataChange Subscription Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/StandardDataChangeSubscription |
| Enhanced DataChange Subscription Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/EnhancedDataChangeSubscription |
| Durable Subscription Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/DurableSubscription |
| Data Access Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/DataAccess |
| ComplexType Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/ComplexTypes |
| Standard Event Subscription Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/StandardEventSubscription |
| Address Space Notifier Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/AddressSpaceNotifier |
| A & C Base Condition Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/ACBaseCondition |

| Profile | Related Category | URI |
|--|------------------|---|
| A & C Address Space Instance Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/ACAddressSpaceInstance |
| A & C Refresh2 Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/ACRefresh2 |
| A & C Enable Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/ACEnable |
| A & C Alarm Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/ACAlarm |
| A & C Acknowledgeable Alarm Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/ACAckAlarm |
| A & C Exclusive Alarming Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/ACExclusiveAlarming |
| A & C Non-Exclusive Alarming Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/ACNon-ExclusiveAlarming |
| A & C Previous Instances Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/ACPreviousInstances |
| A & C Dialog Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/ACDialog |
| A & C CertificateExpiration Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/ACCertificateExpiration |
| A & E Wrapper Facet | Server | http://opcfoundation.org/UA-Profile/Server/AEWrapper |
| Method Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/Methods |
| Auditing Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/Auditing |
| Node Management Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/NodeManagement |
| Client Redundancy Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/ClientRedundancy |
| Redundancy Transparent Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/TransparentRedundancy |
| Redundancy Visible Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/VisibleRedundancy |
| Historical Raw Data Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/HistoricalRawData |
| Historical Aggregate Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/AggregateHistorical |
| Historical Access Structured Data Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/HistoricalStructuredData |
| Historical Data AtTime Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/HistoricalDataAtTime |
| Historical Access Modified Data Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/HistoricalModifiedData |
| Historical Annotation Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/HistoricalAnnotation |
| Historical Data Update Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/HistoricalDataUpdate |
| Historical Data Replace Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/HistoricalDataReplace |
| Historical Data Insert Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/HistoricalDataInsert |
| Historical Data Delete Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/HistoricalDataDelete |
| Base Historical Event Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/BaseHistoricalEvent |
| Historical Event Update Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/HistoricalEventUpdate |
| Historical Event Replace Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/HistoricalEventReplace |
| Historical Event Insert Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/HistoricalEventInsert |
| Historical Event Delete Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/HistoricalEventDelete |

| Profile | Related Category | URI |
|--|------------------|---|
| Aggregate Subscription Server Facet | Server | http://opcfoundation.org/UA-Profile/Server/AggregateSubscription |
| Nano Embedded Device Server Profile | Server | http://opcfoundation.org/UA-Profile/Server/NanoEmbeddedDevice |
| Micro Embedded Device Server Profile | Server | http://opcfoundation.org/UA-Profile/Server/MicroEmbeddedDevice |
| Embedded UA Server Profile | Server | http://opcfoundation.org/UA-Profile/Server/EmbeddedUA |
| Standard UA Server Profile | Server | http://opcfoundation.org/UA-Profile/Server/StandardUA |
| Global Discovery Server Profile | Server | http://opcfoundation.org/UA-Profile/Server/GlobalDiscovery |
| Global Discovery and Certificate Management Server Profile | Server | http://opcfoundation.org/UA-Profile/Server/GlobalDiscoveryAndCertificateManagement |
| Core Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/Core |
| Request State Change Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/RequestStateChange |
| Global Certificate Management Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/GlobalCertificateManagement |
| Base Client Behaviour Facet | Client | http://opcfoundation.org/UA-Profile/Client/Behaviour |
| Discovery Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/Discovery |
| Subnet Discovery Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/SubnetDiscovery |
| Global Discovery Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/GlobalDiscovery |
| AddressSpace Lookup Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/AddressSpaceLookup |
| Entry-Level Support 2015 Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/Entry-LevelSupport2015 |
| Multi-Server Client Connection Facet | Client | http://opcfoundation.org/UA-Profile/Client/MultiServer |
| File Access Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/FileAccess |
| Documentation – Client | Client | http://opcfoundation.org/UA-Profile/Client/Documentation |
| Attribute Read Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/AttributeRead |
| Attribute Write Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/AttributeWrite |
| DataChange Subscriber Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/DataChangeSubscriber |
| Durable Subscription Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/DurableSubscription |
| DataAccess Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/DataAccess |
| Event Subscriber Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/EventSubscriber |
| Base Event Processing Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/BaseEventProcessing |
| Notifier and Source Hierarchy Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/NotifierAndSourceHierarchy |
| A & C Base Condition Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/ACBaseCondition |
| A & C Refresh2 Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/ACRefresh2 |
| A & C Address Space Instance Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/ACAddressSpaceInstance |
| A & C Enable Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/ACEnable |
| A & C Alarm Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/ACAlarm |
| A & C Exclusive Alarming Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/ACExclusiveAlarming |
| A & C Non-Exclusive Alarming Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/ACNon-ExclusiveAlarming |
| A & C Previous Instances Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/ACPreviousInstances |
| A & C Dialog Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/ACDialog |
| A & C CertificateExpiration Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/ACCertificateExpiration |
| A & E Proxy Facet | Client | http://opcfoundation.org/UA-Profile/Client/AEProxy |
| Method Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/Method |
| Auditing Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/Auditing |

| Profile | Related Category | URI |
|--|------------------|---|
| Node Management Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/NodeManagement |
| Advanced Type Programming Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/TypeProgramming |
| Diagnostic Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/Diagnostic |
| Redundant Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/Redundancy |
| Redundancy Switch Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/RedundancySwitch |
| Historical Access Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/HistoricalAccess |
| Historical Annotation Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/HistoricalAnnotation |
| Historical Data AtTime Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/HistoricalAccessAtTime |
| Historical Aggregate Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/HistoricalAccessAggregate |
| Historical Data Update Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/HistoricalUpdateData |
| Historical Data Replace Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/HistoricalReplaceData |
| Historical Data Insert Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/HistoricalInsertData |
| Historical Data Delete Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/HistoricalDeleteData |
| Historical Access Client Server Timestamp Facet | Client | http://opcfoundation.org/UA-Profile/Client/HistoricalServerTimeStamp |
| Historical Access Modified Data Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/HistoricalAccessModifiedData |
| Historical Structured Data AtTime Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/HistoricalAtTimeStructuredData |
| Historical Structured Data Access Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/HistoricalAccessStructuredData |
| Historical Structured Data Modified Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/HistoricalModifiedStructuredData |
| Historical Structured Data Delete Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/HistoricalDeleteStructuredData |
| Historical Structured Data Update Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/HistoricalUpdateStructuredData |
| Historical Structured Data Replace Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/HistoricalReplaceStructuredData |
| Historical Structured Data Insert Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/HistoricalInsertStructuredData |
| Historical Events Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/HistoricalEvents |
| Historical Event Update Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/HistoricalUpdateEvents |
| Historical Event Replace Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/HistoricalReplaceEvents |
| Historical Event Delete Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/HistoricalDeleteEvents |
| Historical Event Insert Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/HistoricalInsertEvents |
| Aggregate Subscriber Client Facet | Client | http://opcfoundation.org/UA-Profile/Client/AggregateSubscriber |
| Global Certificate Management Client Profile | Client | http://opcfoundation.org/UA-Profile/Client/GlobalCertificateManagement |
| Standard UA Client Profile | Client | http://opcfoundation.org/UA-Profile/Client/StandardUA |
| User Token – Anonymous Facet | Security | http://opcfoundation.org/UA-Profile/Security/UserToken/Anonymous |
| User Token – User Name Password Server Facet | Server, Security | http://opcfoundation.org/UA-Profile/Security/UserToken-Server/UserNamePassword |

| Profile | Related Category | URI |
|--|------------------|---|
| User Token – X509 Certificate Server Facet | Server, Security | http://opcfoundation.org/UA-Profile/Security/UserToken-Server/X509Certificate |
| User Token – Issued Token Server Facet | Server, Security | http://opcfoundation.org/UA-Profile/Security/UserToken-Server/IssuedToken |
| User Token – Issued Token Windows Server Facet | Server, Security | http://opcfoundation.org/UA-Profile/Security/UserToken-Server/IssuedTokenWindows |
| User Token – User Name Password Client Facet | Client, Security | http://opcfoundation.org/UA-Profile/Security/UserToken-Client/UserNamePassword |
| User Token – X509 Certificate Client Facet | Client, Security | http://opcfoundation.org/UA-Profile/Security/UserToken-Client/X509Certificate |
| User Token – Issued Token Client Facet | Client, Security | http://opcfoundation.org/UA-Profile/Security/UserToken-Client/IssuedToken |
| User Token – Issued Token Windows Client Facet | Client, Security | http://opcfoundation.org/UA-Profile/Security/UserToken-Client/IssuedTokenWindows |
| UA-TCP UA-SC UA Binary | Transport | http://opcfoundation.org/UA-Profile/Transport/uatcp-uasc-uabinary |
| HTTPS UA Binary | Transport | http://opcfoundation.org/UA-Profile/Transport/https-uabinary |
| HTTPS UA XML | Transport | http://opcfoundation.org/UA-Profile/Transport/https-uasoapxml |
| Security User Access Control Full | Security, Server | http://opcfoundation.org/UA-Profile/Security/UserAccessFull |
| Security User Access Control Base | Security, Server | http://opcfoundation.org/UA-Profile/Security/UserAccessBase |
| Security Time Synchronization | Security | http://opcfoundation.org/UA-Profile/Security/TimeSync |
| Best Practice – Audit Events | Security, Server | http://opcfoundation.org/UA-Profile/Security/BestPracticeAuditEvents |
| Best Practice – Alarm Handling | Security, Server | http://opcfoundation.org/UA-Profile/Security/BestPracticeAlarmHandling |
| Best Practice – Program Access | Security, Server | http://opcfoundation.org/UA-Profile/Security/BestPracticeProgramAccess |
| Best Practice – Random Numbers | Security | http://opcfoundation.org/UA-Profile/Security/BestPracticeRandomNumbers |
| Best Practice – Timeouts | Security | http://opcfoundation.org/UA-Profile/Security/BestPracticeTimeouts |
| Best Practice – Administrative Access | Security | http://opcfoundation.org/UA-Profile/Security/BestPracticeAdministrativeAccess |
| Best Practice – Strict Message Handling | Security, Server | http://opcfoundation.org/UA-Profile/Security/BestPracticeStrictMessage |
| Best Practice – Alarm Handling Client | Client, Security | http://opcfoundation.org/UA-Profile/Security/BestPracticeAlarmHandlingClient |
| Best Practice – Audit Events Client | Client, Security | http://opcfoundation.org/UA-Profile/Security/BestPracticeAuditEventsClient |
| SecurityPolicy – None | Security | http://opcfoundation.org/UA/SecurityPolicy#None |
| SecurityPolicy – Basic128Rsa15 | Security | http://opcfoundation.org/UA/SecurityPolicy#Basic128Rsa15 |
| SecurityPolicy – Basic256 | Security | http://opcfoundation.org/UA/SecurityPolicy#Basic256 |
| SecurityPolicy – Basic256Sha256 | Security | http://opcfoundation.org/UA/SecurityPolicy#Basic256Sha256 |
| TransportSecurity – TLS 1.2 | Security | http://opcfoundation.org/UA-Profile/TransportSecurity/TLS-1-2 |
| TransportSecurity – TLS 1.2 with PFS | Security | http://opcfoundation.org/UA-Profile/TransportSecurity/TLS-1-2-PFS |

The contents of each of the listed *Profiles* will be described in a tabular form in a separate section. Each table may contain references to additional *Profiles* and or *ConformanceUnits*. If a *Profile* is referenced it means that it is completely included. The *ConformanceUnits* are referenced using their name and conformance group. For the details of the *ConformanceUnit* the reader should examine the *ConformanceUnit* details in the appropriate conformance group section.

6.3 Conventions for Profile definitions

Profiles have the following naming conventions:

- *Profiles* intended for OPC UA *Servers* contain the term *Server* in their titles,
- *Profiles* intended for OPC UA *Clients* contain the term *Client* in their titles
- The term *Facet* in the title of a *Profile* indicates that this *Profile* is expected to be part of another larger *Profile* or concerns a specific aspect of OPC UA. *Profiles* with the term *Facet* in their title are expected to be combined with other *Profiles* to define the complete functionality of an OPC UA *Server* or *Client*.

6.4 Applications

A vendor that is developing a UA application, whether it is a *Server* application or a *Client* application, shall review the list of available *Profiles*. From this list the vendor shall select the *Profiles* that include the functionality required by the application. Typically this will be multiple *Profiles*. Conformance to a single *Profile* may not yield a complete application. In most cases multiple *Profiles* are needed to yield a useful application. All *Servers* and *Clients* shall support at least a core *Profile* (Core *Server Facet* or Core *Client Facet*) and at least one Transport *Profile*

For example an HMI *Client* application may choose to support the “Core *Client Facet*”, the “UA-TCP UA-SC UA Binary” *Profile*, the “Data Access *Client Facet*”, the “DataChange Subscriber *Client Facet*” and the “Attribute Write *Client Facet*”. If the *Client* is to be *TestLab* tested then it would also support “Base *Client Behaviour*” *Profile*. This list of *Profiles* would allow the *Client* to communicate with an OPC UA *Server* using UA-TCP/UA Security/UA binary. It would be able to subscribe for data, write to data and would support the DA data model. It would also follow the best practice guideline for behaviour.

Figure 2 illustrates the *Profile* hierarchy that this application may contain: This figure is only an illustration and the represented *Profiles* may change.

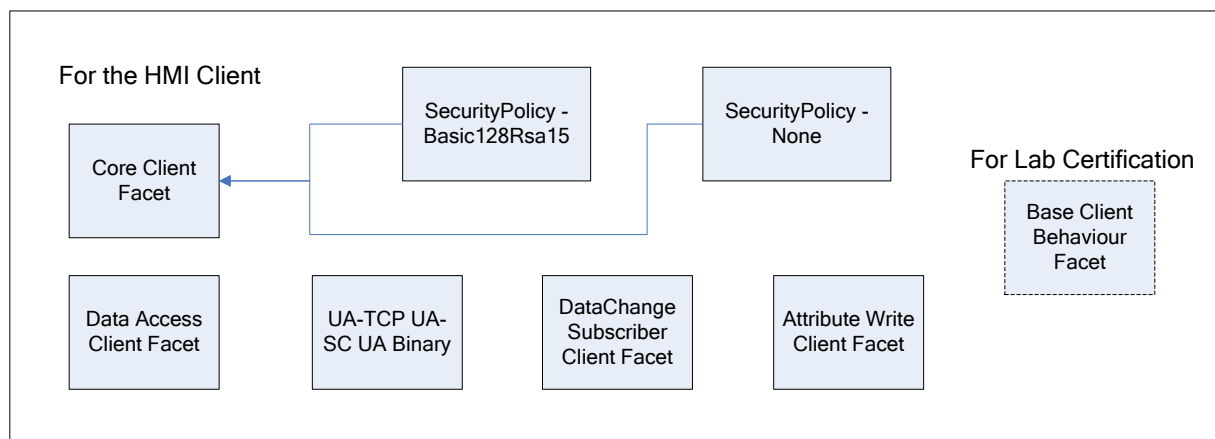


Figure 2 – HMI Client sample

All *Clients* should take into account the types of *Servers* and *Server Profiles* that they are targeted to support. Some *Servers* might not support *Subscriptions* and *Clients* should be able to fall back to Read *Services*.

A special case is a generic *Client* that is designed to communicate with a large number of *Servers* and therefore able to perform a broad range of functionality. “Standard UA *Client Profile*” has been defined for this kind of *Clients*.

Many *Clients*, however, will be specialized and do not need all of the functionality in the “Standard UA *Client Profile*” and thus would only support the limited set of functionality they require. A trend *Client*, for example, would only need functionality to subscribe to or read data.

Another example is an embedded device OPC UA *Server* application that may choose to support “Embedded UA *Server*” *Profile* and the “DataAccess *Server Facet*” *Profile*. This device would be a resource constrained device that would support UA-TCP, UA-Security, UA Binary encoding, data subscriptions and the DA data model. It may not support the optional attribute write. Figure 3 illustrates the hierarchy that this application may contain: This figure is just an illustration and the represented *Profiles* may change.

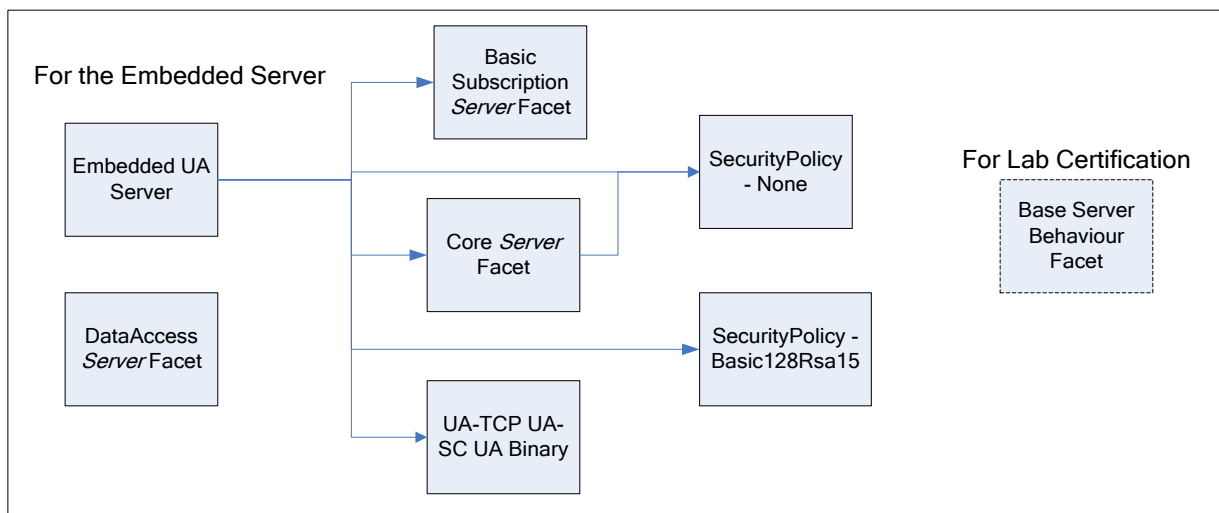


Figure 3 – Embedded Server sample

Another simple system *Server* application may choose to support: “Standard UA *Server*” *Profile* and the “DataAccess *Server Facet*” *Profile*. If the *Server* is to be lab tested then it would also support “Base Server Behaviour” *Profile*. This device would be a mid-level OPC UA *Server* that would support all that the embedded *Server* in the previous example supported and it would add support for an enhance level of the subscription service and support for writes. Figure 4 illustrates the hierarchy that this application may contain: This figure is just an illustration and the represented *Profile* may change.

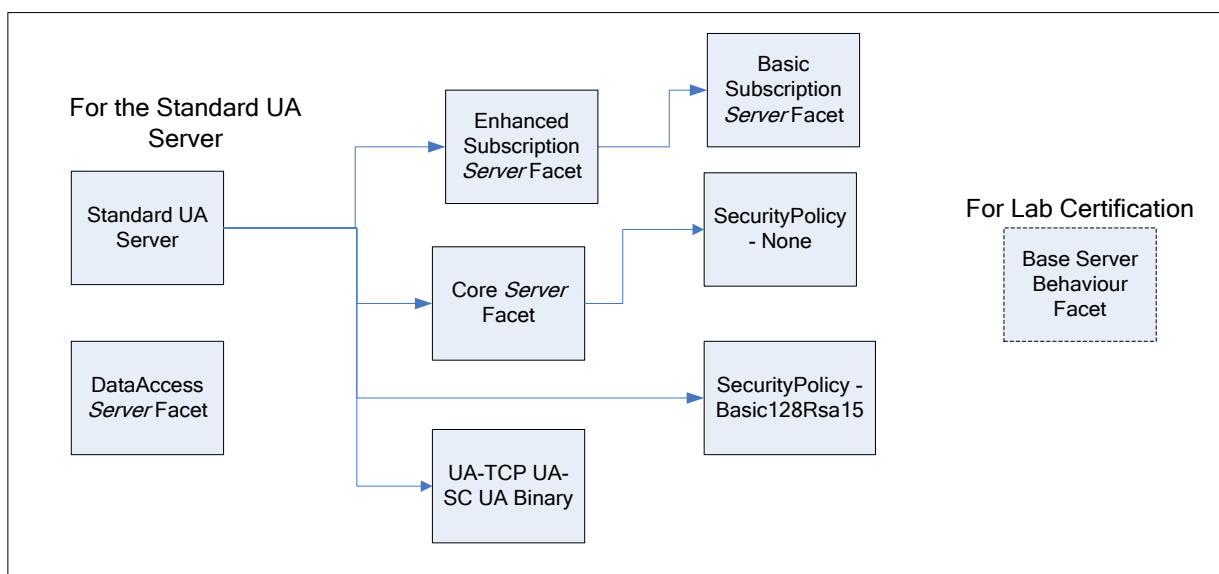


Figure 4 – Standard UA Server sample

If the example HMI *Client* were to connect to either of the example *Servers*, it may have to adjust its behaviour based on the *Profile* reported by the respective *Servers*. If the HMI *Client* were communicating with the embedded device it would not be able to perform any write

operations. It may also have to limit the number of subscriptions or sessions based on the performance limits of the *Server*. If the HMI *Client* is connected to the Standard *Server* it would be able to open additional windows, have higher limits on performance related items and it would be able to allow writes.

6.5 Profile tables

6.5.1 Introduction

All subclauses in 6.5 starting with 6.5.2 describe *Profiles* in a tabular format.

Each table contains three columns. The first column is a description of the conformance group that the *ConformanceUnit* is part of. This allows the reader to easily find the *ConformanceUnit*. This column may also state “*Profile*” in which case the listed item is not a *ConformanceUnit*, but an included *Profile*. The second column is a brief description of the *ConformanceUnit* or included *Profile*. The last column indicates if the *ConformanceUnit* is optional or required.

6.5.2 Core Server Facet

Table 24 describes the details of the Core *Server Facet*. This Facet defines the core functionality required for any UA *Server* implementation. The core functionality includes the ability to discover endpoints, establish secure communication channels, create sessions, browse the *AddressSpace* and read and/or write to attributes of nodes.

The key requirements are: Support for a single session, support for the *Server* and *Server Capabilities Object*, all mandatory *Attributes* for *Nodes* in the *AddressSpace*, and authentication with *UserName* and *Password*.

This Facet has been extended with additional Base Information *ConformanceUnits*. They are optional to provide backward compatibility. In the future the *ConformanceUnit* “Base Info Server Capabilities” will become required, and so it is highly recommended that all *Servers* support it. For broad applicability, it is recommended that *Servers* support multiple transport and security *Profiles*.

Table 24 – Core Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|---------------------|--|----------|
| Profile | SecurityPolicy – None | False |
| Profile | User Token – User Name Password Server Facet | False |
| Address Space Model | Address Space Base | False |
| Attribute Services | Attribute Read | False |
| Attribute Services | Attribute Write Index | True |
| Attribute Services | Attribute Write Values | True |
| Base Information | Base Info Core Structure | False |
| Base Information | Base Info OptionSet | True |
| Base Information | Base Info Placeholder Modelling Rules | True |
| Base Information | Base Info Server Capabilities | True |
| Base Information | Base Info ValueAsText | True |
| Discovery Services | Discovery Find Servers Self | False |
| Discovery Services | Discovery Get Endpoints | False |
| Security | Security – No Application Authentication | True |
| Security | Security Administration | True |
| Session Services | Session Base | False |
| Session Services | Session General Service Behaviour | False |
| Session Services | Session Minimum 1 | False |
| View Services | View Basic | False |
| View Services | View Minimum Continuation Point 01 | False |
| View Services | View RegisterNodes | False |
| View Services | View TranslateBrowsePath | False |

6.5.3 Global Certificate Management Server Facet

Table 25 describes the details of the Global Certificate Management Server Facet. This Facet defines the capability to interact with a *Global Certificate Management Server* to obtain an initial or renewed *Certificate* and *Trust Lists*.

Table 25 – Global Certificate Management Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|----------|--|----------|
| Security | Pull Model for Global Certificate and TrustList Management | True |
| Security | Push Model for Global Certificate and TrustList Management | True |
| Security | Pull or Push Model | False |

6.5.4 Subnet Discovery Server Facet

Table 26 describes the details of the Subnet Discovery Server Facet. Support of this Facet enables discovery of the *Server* on a subnet using mDNS. This functionality is only applicable when *Servers* do not register with an LDS.

Table 26 – Subnet Discovery Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|--------------------|--|----------|
| Discovery Services | Discovery Server Announcement using mDNS | False |

6.5.5 Base Server Behaviour Facet

Table 27 describes the details of the Base Server Behaviour Facet. This Facet defines best practices for the configuration and management of *Servers* when they are deployed in a production environment. It provides the ability to enable or disable certain protocols, to set the security level and to configure the *Discovery Server* and specify where this *Server* shall be registered.

Table 27 – Base Server Behaviour Facet

| Group | Conformance Unit / Profile Title | Optional |
|-----------------------|--------------------------------------|----------|
| Discovery Services | Discovery Configuration | False |
| Protocol and Encoding | Protocol Configuration | False |
| Security | Security Administration | False |
| Security | Security Administration – XML Schema | False |
| Security | Security Certificate Administration | False |

6.5.6 Request State Change Server Facet

Table 28 describes the details of the Request State Change Server Facet. This Facet specifies the support of the RequestServerStateChange Method.

Table 28 – Request State Change Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|------------------|------------------------------------|----------|
| Base Information | Base Info RequestServerStateChange | False |

6.5.7 Attribute WriteMask Server Facet

Table 29 describes the details of the Attribute WriteMask Server Facet. This Facet defines the capability to update characteristics of individual *Nodes* in the *AddressSpace* by allowing writing to *Node Attributes*. It requires support for authenticating user access as well as providing information related to access rights in the *AddressSpace* and actually restricting the access rights as described.

Table 29 – Attribute WriteMask Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|---------------------|--|----------|
| <i>Profile</i> | Security User Access Control Base | False |
| Address Space Model | Address Space UserWriteMask | False |
| Address Space Model | Address Space UserWriteMask Multilevel | True |
| Address Space Model | Address Space WriteMask | False |

6.5.8 File Access Server Facet

Table 30 describes the details of the File Access Server Facet. This Facet specifies the support of exposing File information via the defined FileType. This includes reading of file as well as optionally writing of file data.

Table 30 –File Access Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|------------------|----------------------------------|----------|
| Base Information | Base Info FileType Base | False |
| Base Information | Base Info FileType Write | True |

6.5.9 Documentation Server Facet

Table 31 describes the details of the Documentation Server Facet. This Facet defines a list of user documentation that a server application should provide.

Table 31 – Documentation Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|---------------|--|----------|
| Miscellaneous | Documentation – Installation | False |
| Miscellaneous | Documentation – Multiple Languages | True |
| Miscellaneous | Documentation – On-line | True |
| Miscellaneous | Documentation – Supported Profiles | True |
| Miscellaneous | Documentation – Trouble Shooting Guide | True |
| Miscellaneous | Documentation – Users Guide | False |

6.5.10 Embedded DataChange Subscription Server Facet

Table 32 describes the details of the Embedded DataChange *Subscription Server Facet*. This Facet specifies the minimum level of support for data change notifications within subscriptions. It includes limits which minimize memory and processing overhead required to implement the Facet. This Facet includes functionality to create, modify and delete Subscriptions and to add, modify and remove Monitored Items. As a minimum for each *Session*, Servers shall support one *Subscription* with up to two items. In addition, support for two parallel Publish requests is required. This Facet is geared for a platform such as the one provided by the Micro Embedded Device *Server Profile* in which memory is limited and needs to be managed.

Table 32 – Embedded DataChange Subscription Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|-------------------------|-------------------------------------|----------|
| Monitored Item Services | Monitor Basic | False |
| Monitored Item Services | Monitor Items 2 | False |
| Monitored Item Services | Monitor QueueSize_1 | False |
| Monitored Item Services | Monitor Value Change | False |
| Subscription Services | Subscription Basic | False |
| Subscription Services | Subscription Minimum 1 | False |
| Subscription Services | Subscription Publish Discard Policy | False |
| Subscription Services | Subscription Publish Min 02 | False |

6.5.11 Standard DataChange Subscription Server Facet

Table 33 describes the details of the Standard DataChange *Subscription Server Facet*. This Facet specifies the standard support of subscribing to data changes. This Facet extends

features and limits defined by the Embedded Data Change *Subscription* Facet. As a minimum, *Servers* shall support 2 Subscriptions with at least 100 items for at least half of the required Sessions. The 100 items shall be supported for at least half of the required Subscriptions. Queuing with up to two queued entries is required. Support of five parallel Publish requests per *Session* is required. This Facet also requires the support of the triggering service. This Facet has been updated to include optional *ConformanceUnits* to allow for backward compatibility. These optional *ConformanceUnits* are highly recommended, in that in a future release they will be made mandatory.

Table 33 – Standard DataChange Subscription Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|-------------------------|---|-----------------|
| <i>Profile</i> | Embedded DataChange Subscription Server Facet | False |
| Base Information | Base Info GetMonitoredItems Method | True |
| Method Services | Method Call | True |
| Monitored Item Services | Monitor Items 10 | False |
| Monitored Item Services | Monitor Items 100 | False |
| Monitored Item Services | Monitor MinQueueSize_02 | False |
| Monitored Item Services | Monitor Triggering | False |
| Monitored Item Services | Monitored Items Deadband Filter | False |
| Subscription Services | Subscription Minimum 02 | False |
| Subscription Services | Subscription Publish Min 05 | False |

6.5.12 Enhanced DataChange Subscription Server Facet

Table 34 describes the details of the Enhanced DataChange *Subscription Server* Facet. This Facet specifies an enhanced support of subscribing to data changes. It is part of the Standard UA *Server Profile*. This Facet increases the limits defined by the Standard Data Change *Subscription* Facet.

Table 34 – Enhanced DataChange Subscription Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|-------------------------|---|----------|
| <i>Profile</i> | Standard DataChange Subscription Server Facet | False |
| Monitored Item Services | Monitor Items 500 | False |
| Monitored Item Services | Monitor MinQueueSize_05 | False |
| Subscription Services | Subscription Minimum 05 | False |
| Subscription Services | Subscription Publish Min 10 | False |

6.5.13 Durable Subscription Server Facet

Table 35 describes the details of the Durable *Subscription Server* Facet. This Facet specifies support of durable storage of data and events even when *Clients* are disconnected. This Facet implies support of any of the DataChange or Event Subscription Facets.

Table 35 – Durable Subscription Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|-----------------------|----------------------------------|----------|
| Subscription Services | Subscription Durable | False |

6.5.14 Data Access Server Facet

Table 36 describes the details of the Data Access *Server* Facet. This Facet specifies the support for an *Information Model* used to provide industrial automation data. This model defines standard structures for analog and discrete data items and their quality of service. This Facet extends the Core *Server* Facet which includes support of the basic *AddressSpace* behaviour.

Table 36 – Data Access Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|-------------|----------------------------------|----------|
| Data Access | Data Access AnalogItems | True |
| Data Access | Data Access ArrayItemType | True |
| Data Access | Data Access Complex Number | True |
| Data Access | Data Access DataItems | False |
| Data Access | Data Access DoubleComplex Number | True |
| Data Access | Data Access MultiState | True |
| Data Access | Data Access PercentDeadband | True |
| Data Access | Data Access Semantic Changes | True |
| Data Access | Data Access TwoState | True |

6.5.15 ComplexType Server Facet

Table 37 describes the details of the ComplexType *Server* Facet. This Facet extends the Core *Server* Facet to include *Variables* with *Complex Data*, i.e. data that are composed of multiple elements such as a structure and where the individual elements are exposed as component variables. Support of this Facet requires the implementation of StructuredDataTypes and *Variables* that make use of these DataTypes. The Read, Write and Subscriptions service set shall support the encoding and decoding of these StructuredDataTypes. As an option the *Server* can also support alternate encodings, such as an XML encoding when the binary protocol is currently used and vice-versa.

Table 37 – ComplexType Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|-------------------------|----------------------------------|----------|
| Address Space Model | Address Space Complex DataTypes | False |
| Attribute Services | Attribute Alternate Encoding | True |
| Attribute Services | Attribute Read Complex | False |
| Attribute Services | Attribute Write Complex | False |
| Monitored Item Services | Monitor Alternate Encoding | True |

6.5.16 Standard Event Subscription Server Facet

Table 38 describes the details of the Standard *Event Subscription Server* Facet. This Facet specifies the standard support for subscribing to events and is intended to supplement any of the *FullFeatured Profiles*. Support of this Facet requires the implementation of *Event Types* representing the Events that the *Server* can report and their specific fields. It also requires at least the *Server Object* to have the *EventNotifier Attribute* set. It includes the *Services* to Create, Modify and Delete *Subscriptions* and to Add, Modify and Remove Monitored Items for *Object Nodes* with an “*EventNotifier Attribute*”. Creating a monitoring item may include a filter that includes SimpleAttribute FilterOperands and a select list of Operators. The operators include: Equals, IsNull, GreaterThan, LessThan, GreaterThanOrEqual, LessThanOrEqual, Like, Not, Between, InList, And, Or, Cast, BitwiseAnd, BitwiseOr and TypeOf. Support of more complex filters is optional.

This Facet has been updated to include several optional Base Information *ConformanceUnits*. These *ConformanceUnits* are optional to allow for backward compatibility, in the future these optional *ConformanceUnits* will become required, and so it is highly recommended that all servers support them.

Table 38 – Standard Event Subscription Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|-------------------------|---|----------|
| Address Space Model | Address Space Events | False |
| Base Information | Base Info EventQueueOverflowEventType | True |
| Base Information | Base Info Progress Events | True |
| Base Information | Base Info SemanticChange | True |
| Base Information | Base Info System Status | True |
| Base Information | Base Info System Status underlying system | True |
| Base Information | Base Info Device Failure | True |
| Monitored Item Services | Monitor Basic | False |
| Monitored Item Services | Monitor Complex Event Filter | True |
| Monitored Item Services | Monitor Events | False |
| Monitored Item Services | Monitor Items 10 | False |
| Monitored Item Services | Monitor QueueSize_ServerMax | False |
| Subscription Services | Subscription Basic | False |
| Subscription Services | Subscription Minimum 02 | False |
| Subscription Services | Subscription Publish Discard Policy | False |
| Subscription Services | Subscription Publish Min 05 | False |

6.5.17 Address Space Notifier Server Facet

Table 39 describes the details of the Address Space Notifier *Server* Facet. This Facet requires the support of a hierarchy of *Object Nodes* that are notifiers and *Nodes* that are event sources. The hierarchy is commonly used as a way to organize a plant into areas that can be managed by different operators.

Table 39 – Address Space Notifier Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|---------------------|----------------------------------|----------|
| Address Space Model | Address Space Notifier Hierarchy | False |
| Address Space Model | Address Space Source Hierarchy | False |

6.5.18 A & C Base Condition Server Facet

Table 40 describes the details of the A & C Base Condition *Server* Facet. This Facet requires basic support for *Conditions*. Information about *Conditions* is provided through *Event* notifications and thus this Facet builds upon the Standard *Event Subscription Server* Facet. *Conditions* that are in an “interesting” state (as defined by the *Server*) can be refreshed using the *Refresh Method*, which requires support for the *Method Server* Facet. Optionally the server may also provide support for *Condition* classes

Table 40 – A & C Base Condition Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|-----------------------|--|----------|
| Profile | Method Server Facet | False |
| Profile | Standard Event Subscription Server Facet | False |
| Alarms and Conditions | A & C Basic | False |
| Alarms and Conditions | A & C ConditionClasses | True |
| Alarms and Conditions | A & C Refresh | False |

6.5.19 A & C Refresh2 Server Facet

Table 41 describes the details of the A & C Refresh2 Server Facet. This Facet enhances the A & C Base Condition Server Facet with support of the *ConditionRefresh2 Method*.

Table 41 – A & C Refresh2 Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|-----------------------|-----------------------------------|----------|
| Profile | A & C Base Condition Server Facet | False |
| Alarms and Conditions | A & C Refresh2 | False |

6.5.20 A & C Address Space Instance Server Facet

Table 42 describes the details of the A & C Address Space Instance Server Facet. This Facet specifies the support required for a *Server* to expose *Alarms* and *Conditions* in its *AddressSpace*. This includes the A & C *AddressSpace* information model.

Table 42 – A & C Address Space Instance Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|-----------------------|----------------------------------|----------|
| Alarms and Conditions | A & C Instances | False |

6.5.21 A & C Enable Server Facet

Table 43 describes the details of the A & C Enable Server Facet. This Facet requires the enabling and disabling of *Conditions*. This facet builds upon the A&C Base Condition Server Facet. Enabling and disabling also requires that instances of these *ConditionTypes* exist in the *AddressSpace* since the enable *Method* can only be invoked on an instance of the *Condition*

Table 43 – A & C Enable Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|-----------------------|-----------------------------------|----------|
| Profile | A & C Base Condition Server Facet | False |
| Alarms and Conditions | A & C Enable | False |
| Alarms and Conditions | A & C Instances | False |

6.5.22 A & C Alarm Server Facet

Table 44 describes the details of the A & C Alarm Server Facet. This Facet requires support for *Alarms*. *Alarms* extend the *ConditionType* by adding an Active state which indicates when something in the system requires attention by an Operator. This Facet builds upon the A&C Base Condition Server Facet. This facet requires that discrete *AlarmTypes* be supported, it also allows for optional support of shelving, alarm comments and other discrete *AlarmTypes* such as Trip or Off-Normal.

Table 44 – A & C Alarm Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|-----------------------|-----------------------------------|----------|
| Profile | A & C Base Condition Server Facet | False |
| Alarms and Conditions | A & C Alarm | False |
| Alarms and Conditions | A & C Comment | True |
| Alarms and Conditions | A & C Discrete | False |
| Alarms and Conditions | A & C OffNormal | True |

| Group | Conformance Unit / Profile Title | Optional |
|-----------------------|----------------------------------|----------|
| Alarms and Conditions | A & C SystemOffNormal | True |
| Alarms and Conditions | A & C Shelving | True |
| Alarms and Conditions | A & C Trip | True |

6.5.23 A & C Acknowledgeable Alarm Server Facet

Table 45 describes the details of the A & C Acknowledgeable *Alarm Server Facet*. This Facet requires support for Acknowledgement of active *Alarms*. This Facet builds upon the A & C *Alarm Server Facet*. Acknowledgement requires support of the Acknowledge *Method* and the Acknowledged state. Support of the Confirmed state and the Confirm *Method* is optional.

Table 45 – A & C Acknowledgeable Alarm Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|-----------------------|----------------------------------|----------|
| Profile | A & C Alarm Server Facet | False |
| Alarms and Conditions | A & C Acknowledge | False |
| Alarms and Conditions | A & C Confirm | True |

6.5.24 A & C Exclusive Alarming Server Facet

Table 46 describes the details of the A & C Exclusive Alarming *Server Facet*. This Facet requires support for *Alarms* with multiple sub-states that identify different limit *Conditions*. This facet builds upon the A&C *Alarm Server Facet*. The term exclusive means only one sub-state can be active at a time. For example, a temperature exceeds the HighHigh limit the associated exclusive LevelAlarm will be in the HighHigh sub-state and not in the High sub-state. This Facet requires that a *Server* support at least one of the optional *Alarm* models: Limit, RateOfChange or Deviation.

Table 46 – A & C Exclusive Alarming Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|-----------------------|----------------------------------|----------|
| Profile | A & C <i>Alarm Server Facet</i> | False |
| Alarms and Conditions | A & C Exclusive Deviation | True |
| Alarms and Conditions | A & C Exclusive Level | True |
| Alarms and Conditions | A & C Exclusive Limit | False |
| Alarms and Conditions | A & C Exclusive RateOfChange | True |

6.5.25 A & C Non-Exclusive Alarming Server Facet

Table 47 describes the details of the A & C Non-Exclusive Alarming *Server Facet*. This Facet requires support for *Alarms* with multiple sub-states that identify different limit *Conditions*. This Facet builds upon the A&C *Alarm Server Facet*. The term non-exclusive means more than one sub-state can be active at a time. For example, if a temperature exceeds the HighHigh limit the associated non-exclusive LevelAlarm will be in both the High and the HighHigh sub-state. This Facet requires that a server support at least one of the optional alarm models: Limit, RateOfChange or Deviation.

Table 47 – A & C Non-Exclusive Alarming Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|-----------------------|----------------------------------|----------|
| Profile | A & C Alarm Server Facet | False |
| Alarms and Conditions | A & C Non-Exclusive Deviation | True |
| Alarms and Conditions | A & C Non-Exclusive Level | True |
| Alarms and Conditions | A & C Non-Exclusive Limit | False |
| Alarms and Conditions | A & C Non-Exclusive RateOfChange | True |

6.5.26 A & C Previous Instances Server Facet

Table 48 describes the details of the A & C Previous Instances *Server Facet*. This Facet requires support for *Conditions* with previous states that still require action on the part of the operator. This facet builds upon the A&C Base Condition *Server Facet*. A common use case for this Facet

is a safety critical system that requires that all *Alarms* be acknowledged even if the original problem goes away and the *Alarm* returns to the inactive state. In these cases, the previous state with active *Alarm* is still reported by the *Server* until the Operator acknowledges it. When a *Condition* has previous states it will produce events with different Branch identifiers. When previous state no longer needs attention the branch will disappear.

Table 48 – A & C Previous Instances Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|-----------------------|-----------------------------------|----------|
| Profile | A & C Base Condition Server Facet | False |
| Alarms and Conditions | A & C Branch | False |

6.5.27 A & C Dialog Server Facet

Table 49 describes the details of the A & C Dialog *Server* Facet. This Facet requires support of Dialog *Conditions*. This Facet builds upon the A & C Base Condition Server Facet Dialogs are ConditionTypes used to request user input. They are typically used when a *Server* has entered some state that requires intervention by a *Client*. For example, a *Server* monitoring a paper machine indicates that a roll of paper has been wound and is ready for inspection. The *Server* would activate a Dialog *Condition* indicating to the user that an inspection is required. Once the inspection has taken place the user responds by informing the *Server* of an accepted or unaccepted inspection allowing the process to continue.

Table 49 – A & C Dialog Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|-----------------------|-----------------------------------|----------|
| Profile | A & C Base Condition Server Facet | False |
| Alarms and Conditions | A & C Dialog | False |

6.5.28 A & C CertificateExpiration Server Facet

Table 50 describes the details of the A & C CertificateExpiration *Server* Facet. This Facet requires support of the *CertificateExpirationAlarmType*. It is used to inform Clients when the *Server's* Certificate is within the defined expiration period.

Table 50 – A & C CertificateExpiration Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|-----------------------|-----------------------------------|----------|
| Profile | A & C Base Condition Server Facet | False |
| Alarms and Conditions | A & C Alarm | False |
| Alarms and Conditions | A & C Comment | True |
| Alarms and Conditions | A & C Shelving | True |
| Alarms and Conditions | A & C Acknowledge | True |
| Alarms and Conditions | A & C Confirm | True |
| Alarms and Conditions | A & C CertificateExpiration | False |

6.5.29 A & E Wrapper Facet

Table 51 describes the details of the A & E Wrapper Facet. This Facet specifies the requirements for a UA *Server* that wraps an OPC *Alarm & Event* (AE) *Server* (COM). This *Profile* identifies the sub-set of the UA *Alarm & Condition* model which is provided by the COM OPC AE specification. It is intended to provide guidance to developers who are creating servers that front-end existing applications. It is important to note that some OPC A&E COM *Servers* may not support all of the functionality provided by an OPC UA A&C server, in these cases similar functionality maybe available via some non-OPC interface. For example if an A&E COM server does not support sending *Alarm* Acknowledgement messages to the system that it is obtaining alarm information from, this functionality may be available via some out of scope features in the underlying *Alarm* system. Another possibility is that the underlying system does not require acknowledgements or automatically acknowledges the alarm.

Table 51 – A & E Wrapper Facet

| Group | Conformance Unit / Profile Title | Optional |
|-------------------------|---|-----------------|
| Address Space Model | Address Space Events | False |
| Address Space Model | Address Space Notifier Hierarchy | False |
| Address Space Model | Address Space Source Hierarchy | False |
| Alarms and Conditions | A & C Acknowledge | False |
| Alarms and Conditions | A & C Alarm | False |
| Alarms and Conditions | A & C Basic | False |
| Alarms and Conditions | A & C ConditionClasses | False |
| Alarms and Conditions | A & C Refresh | False |
| Alarms and Conditions | A & E Wrapper Mapping | False |
| Monitored Item Services | Monitor Basic | False |
| Monitored Item Services | Monitor Complex Event Filter | False |
| Monitored Item Services | Monitor Events | False |
| Monitored Item Services | Monitor Items 2 | False |
| Monitored Item Services | Monitor QueueSize_ServerMax | False |
| Subscription Services | Subscription Basic | False |
| Subscription Services | Subscription Minimum 1 | False |
| Subscription Services | Subscription Publish Discard Policy | False |
| Subscription Services | Subscription Publish Min 02 | False |

6.5.30 Method Server Facet

Table 52 describes the details of the *Method Server* Facet. This Facet specifies the support of *Method* invocation via the Call service. Methods are “lightweight” functions which are similar to the methods of a class found in any object-oriented programming language. A *Method* can have its scope bounded by an owning *Object* or an owning *ObjectType*. Methods with an *ObjectType* as their scope are similar to static methods in a class.

Table 52 – Method Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|---------------------|----------------------------------|----------|
| Address Space Model | Address Space Method | False |
| Method Services | Method Call | False |

6.5.31 Auditing Server Facet

Table 53 describes the details of the Auditing *Server Facet*. This Facet requires the support of Auditing which includes the Standard *Event Subscription Server Facet*. Support of this Facet requires that Audit Events be produced when a client performs some action to change the state of the server, such as changing the *AddressSpace*, inserting or updating a value etc. The *auditEntryId* passed by the *Client* is a field contained in every Audit *Event* and allows actions to be traced across multiple systems. The Audit *Event* Types and their fields must be exposed in the *Server's AddressSpace*

Table 53 – Auditing Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|----------------|--|----------|
| <i>Profile</i> | Standard Event Subscription Server Facet | False |
| Auditing | Auditing Base | False |

6.5.32 Node Management Server Facet

Table 54 describes the details of the *Node Management Server Facet*. This Facet requires the support of the *Services* that allow the *Client* to add, modify and delete *Nodes* in the *AddressSpace*. These *Services* provide an interface which can be used to configure *Servers*. This means all changes to the *AddressSpace* are expected to persist even after the *Client* has disconnected from the *Server*

Table 54 – Node Management Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|--------------------------|----------------------------------|----------|
| Address Space Model | Address Space Base | False |
| Base Information | Base Info Model Change | False |
| Base Information | Base Info Type System | False |
| Node Management Services | Node Management Add Node | False |
| Node Management Services | Node Management Add Ref | False |
| Node Management Services | Node Management Delete Node | False |
| Node Management Services | Node Management Delete Ref | False |

6.5.33 Client Redundancy Server Facet

Table 55 describes the details of the *Client Redundancy Server Facet*. This Facet defines the *Server* actions that are required for support of redundant *Clients*. Support of this Facet requires the implementation of the *TransferSubscriptions Service* which allows the transfer of Subscriptions from one *Client's Session* to another *Client's Session*.

Table 55 – Client Redundancy Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|-----------------------|----------------------------------|----------|
| Subscription Services | Subscription Transfer | False |

6.5.34 Redundancy Transparent Server Facet

Table 56 describes the details of the *Redundancy Transparent Server Facet*. This Facet requires support for transparent redundancy. If *Servers* implement transparent redundancy then the failover from one *Server* to another is transparent to the *Client* such that the *Client* is

unaware that a failover has occurred; the *Client* does not need to do anything at all to keep data flowing. This type of redundancy is usually a hardware solution.

Table 56 – Redundancy Transparent Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|------------|----------------------------------|----------|
| Redundancy | Redundancy Server Transparent | False |

6.5.35 Redundancy Visible Server Facet

Table 57 describes the details of the Redundancy Visible Server Facet. This Facet specifies the support for non-transparent redundancy. Failover for this type of redundancy requires the *Client* to monitor *Server* status and to switch to a backup *Server* if it detects a failure. The *Server* shall expose the methods of failover it supports (cold, warm or hot). The failover method tells the *Client* what it must do when connecting to a *Server* and when a failure occurs. Cold redundancy requires a *Client* to reconnect to a backup *Server* after the initial *Server* has failed. Warm redundancy allows a *Client* to connect to multiple *Servers*, but only one *Server* will be providing values. In hot redundancy multiple *Servers* are able to provide data and a *Client* can connect to multiple *Servers* for the data.

Table 57 – Redundancy Visible Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|------------|----------------------------------|----------|
| Redundancy | Redundancy Server | False |

6.5.36 Historical Raw Data Server Facet

Table 58 describes the details of the Historical Raw Data Server Facet. This Facet defines the basic functionality when supporting historical data access for raw data.

Table 58 – Historical Raw Data Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|--------------------|--|----------|
| Attribute Services | Attribute Historical Read | False |
| Historical Access | Historical Access Data Max Nodes Read Continuation Point | False |
| Historical Access | Historical Access Read Raw | False |
| Historical Access | Historical Access ServerTimestamp | True |

6.5.37 Historical Aggregate Server Facet

Table 59 describes the details of the Historical Aggregate Server Facet. This Facet indicates that the server supports aggregate processing to produce derived values from raw historical data.

Table 59 – Historical Aggregate Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|------------|------------------------------------|----------|
| Aggregates | Aggregate – AnnotationCount | True |
| Aggregates | Aggregate – Average | True |
| Aggregates | Aggregate – Count | True |
| Aggregates | Aggregate – Custom | True |
| Aggregates | Aggregate – Delta | True |
| Aggregates | Aggregate – DeltaBounds | True |
| Aggregates | Aggregate – DurationBad | True |
| Aggregates | Aggregate – DurationGood | True |
| Aggregates | Aggregate – DurationInStateNonZero | True |
| Aggregates | Aggregate – DurationInStateZero | True |
| Aggregates | Aggregate – End | True |
| Aggregates | Aggregate – EndBound | True |
| Aggregates | Aggregate – Interpolative | True |
| Aggregates | Aggregate – Maximum | True |

| Group | Conformance Unit / Profile Title | Optional |
|--------------------|--|----------|
| Aggregates | Aggregate – Maximum2 | True |
| Aggregates | Aggregate – MaximumActualTime | True |
| Aggregates | Aggregate – MaximumActualTime2 | True |
| Aggregates | Aggregate – Minimum | True |
| Aggregates | Aggregate – Minimum2 | True |
| Aggregates | Aggregate – MinimumActualTime | True |
| Aggregates | Aggregate – MinimumActualTime2 | True |
| Aggregates | Aggregate – NumberOfTransitions | True |
| Aggregates | Aggregate – PercentBad | True |
| Aggregates | Aggregate – PercentGood | True |
| Aggregates | Aggregate – Range | True |
| Aggregates | Aggregate – Range2 | True |
| Aggregates | Aggregate – StandardDeviationPopulation | True |
| Aggregates | Aggregate – StandardDeviationSample | True |
| Aggregates | Aggregate – Start | True |
| Aggregates | Aggregate – StartBound | True |
| Aggregates | Aggregate – TimeAverage | True |
| Aggregates | Aggregate – TimeAverage2 | True |
| Aggregates | Aggregate – Total | True |
| Aggregates | Aggregate – Total2 | True |
| Aggregates | Aggregate – VariancePopulation | True |
| Aggregates | Aggregate – VarianceSample | True |
| Aggregates | Aggregate – WorstQuality | True |
| Aggregates | Aggregate – WorstQuality2 | True |
| Aggregates | Aggregate master configuration | False |
| Aggregates | Aggregate optional configuration | True |
| Attribute Services | Attribute Historical Read | False |
| Historical Access | Historical Access Aggregates | False |
| Historical Access | Historical Access Data Max Nodes Read Continuation Point | False |

6.5.38 Historical Access Structured Data Server Facet

Table 60 describes the details of the Historical Access Structured Data Server Facet. This Facet indicates that the Server supports storage and retrieval of structured values for all supported access types. If a listed access type is supported then the corresponding optional *ConformanceUnit* shall be supported.

Table 60 – Historical Access Structured Data Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|-------------------|---|----------|
| Historical Access | Historical Access Structured Data Delete | True |
| Historical Access | Historical Access Structured Data Insert | True |
| Historical Access | Historical Access Structured Data Read Modified | True |
| Historical Access | Historical Access Structured Data Read Raw | False |
| Historical Access | Historical Access Structured Data Time Instance | True |
| Historical Access | Historical Access Structured Data Update | True |
| Historical Access | Historical Access Structured Data Replace | True |

6.5.39 Historical Data AtTime Server Facet

Table 61 describes the details of the Historical Data AtTime Server Facet. This Facet indicates that the historical Server supports reading data by specifying specific timestamps.

Table 61 – Historical Data AtTime Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|--------------------|----------------------------------|----------|
| Attribute Services | Attribute Historical Read | False |

| Group | Conformance Unit / <i>Profile</i> Title | Optional |
|-------------------|--|----------|
| Historical Access | Historical Access Data Max Nodes Read Continuation Point | False |
| Historical Access | Historical Access Time Instance | False |

6.5.40 Historical Access Modified Data Server Facet

Table 62 describes the details of the Historical Access Modified Data *Server* Facet. This Facet defines support of reading modified historical values (values that where modified or inserted).

Table 62 – Historical Access Modified Data Server Facet

| Group | Conformance Unit / <i>Profile</i> Title | Optional |
|--------------------|---|----------|
| Attribute Services | Attribute Historical Read | False |
| Historical Access | Historical Access Modified Values | False |

6.5.41 Historical Annotation Server Facet

Table 63 describes the details of the Historical Annotation *Server* Facet. This Facet defines support for the storage and retrieval of annotations for historical data.

Table 63 – Historical Annotation Server Facet

| Group | Conformance Unit / <i>Profile</i> Title | Optional |
|--------------------|---|----------|
| Attribute Services | Attribute Historical Read | False |
| Attribute Services | Attribute Historical Update | False |
| Historical Access | Historical Access Annotations | False |

6.5.42 Historical Data Update Server Facet

Table 64 describes the details of the Historical Data Update *Server* Facet. This Facet includes Historical Data Update functionality.

Table 64 – Historical Data Update Server Facet

| Group | Conformance Unit / <i>Profile</i> Title | Optional |
|--------------------|---|----------|
| Attribute Services | Attribute Historical Update | False |
| Historical Access | Historical Access ServerTimestamp | True |
| Historical Access | Historical Access Update Value | False |

6.5.43 Historical Data Replace Server Facet

Table 65 describes the details of the Historical Data Replace *Server* Facet. This Facet includes Historical Data Replace functionality.

Table 65 – Historical Data Replace Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|--------------------|-----------------------------------|----------|
| Attribute Services | Attribute Historical Update | False |
| Historical Access | Historical Access ServerTimestamp | True |
| Historical Access | Historical Access Replace Value | False |

6.5.44 Historical Data Insert Server Facet

Table 66 describes the details of the Historical Data Insert *Server Facet*. This Facet includes Historical Data Insert functionality.

Table 66 – Historical Data Insert Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|--------------------|-----------------------------------|----------|
| Attribute Services | Attribute Historical Update | False |
| Historical Access | Historical Access Insert Value | False |
| Historical Access | Historical Access ServerTimestamp | True |

6.5.45 Historical Data Delete Server Facet

Table 67 describes the details of the Historical Data Delete *Server Facet*. This Facet includes Historical Data Delete functionality.

Table 67 – Historical Data Delete Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|--------------------|----------------------------------|----------|
| Attribute Services | Attribute Historical Update | False |
| Historical Access | Historical Access Delete Value | False |

6.5.46 Base Historical Event Server Facet

Table 68 describes the details of the Base Historical *Event Server Facet*. This Facet defines the server requirements to support basic Historical *Event* functionality, including simple filtering and general access.

Table 68 – Base Historical Event Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|--------------------|--|----------|
| Attribute Services | Attribute Historical Read | False |
| Historical Access | Historical Access Event Max Events Read Continuation Point | False |
| Historical Access | Historical Access Events | False |

6.5.47 Historical Event Update Server Facet

Table 69 describes the details of the Historical *Event Update Server Facet*. This Facet includes Historical *Event* update access functionality.

Table 69 – Historical Event Update Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|--------------------|----------------------------------|----------|
| Attribute Services | Attribute Historical Update | False |
| Historical Access | Historical Access Update Event | False |

6.5.48 Historical Event Replace Server Facet

Table 69 describes the details of the Historical *Event Replace Server Facet*. This Facet includes Historical *Event* replace access functionality.

Table 70 – Historical Event Replace Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|--------------------|----------------------------------|----------|
| Attribute Services | Attribute Historical Update | False |
| Historical Access | Historical Access Replace Event | False |

6.5.49 Historical Event Insert Server Facet

Table 71 describes the details of the Historical *Event* Insert Server Facet. This Facet includes Historical *Event* insert access functionality.

Table 71 – Historical Event Insert Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|--------------------|----------------------------------|----------|
| Attribute Services | Attribute Historical Update | False |
| Historical Access | Historical Access Insert Event | False |

6.5.50 Historical Event Delete Server Facet

Table 72 describes the details of the Historical *Event* Delete Server Facet. This Facet includes Historical *Event* delete access functionality

Table 72 – Historical Event Delete Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|--------------------|----------------------------------|----------|
| Attribute Services | Attribute Historical Update | False |
| Historical Access | Historical Access Delete Event | False |

6.5.51 Aggregate Subscription Server Facet

Table 73 describes the details of the Aggregate *Subscription* Server Facet. This Facet defines the handling of the aggregate filter when subscribing for *Attribute* values.

Table 73 – Aggregate Subscription Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|------------|---|----------|
| Profile | Standard DataChange Subscription Server Facet | False |
| Aggregates | Aggregate Subscription – AnnotationCount | True |
| Aggregates | Aggregate Subscription – Average | True |
| Aggregates | Aggregate Subscription – Count | True |
| Aggregates | Aggregate Subscription – Custom | True |
| Aggregates | Aggregate Subscription – Delta | True |
| Aggregates | Aggregate Subscription – DeltaBounds | True |
| Aggregates | Aggregate Subscription – DurationBad | True |
| Aggregates | Aggregate Subscription – DurationGood | True |
| Aggregates | Aggregate Subscription – DurationInStateNonZero | True |
| Aggregates | Aggregate Subscription – DurationInStateZero | True |
| Aggregates | Aggregate Subscription – End | True |
| Aggregates | Aggregate Subscription – EndBound | True |
| Aggregates | Aggregate Subscription – Filter | False |
| Aggregates | Aggregate Subscription – Interpolative | True |
| Aggregates | Aggregate Subscription – Maximum | True |
| Aggregates | Aggregate Subscription – Maximum2 | True |
| Aggregates | Aggregate Subscription – MaximumActualTime | True |
| Aggregates | Aggregate Subscription – MaximumActualTime2 | True |
| Aggregates | Aggregate Subscription – Minimum | True |
| Aggregates | Aggregate Subscription – Minimum2 | True |
| Aggregates | Aggregate Subscription – MinimumActualTime | True |
| Aggregates | Aggregate Subscription – MinimumActualTime2 | True |
| Aggregates | Aggregate Subscription – NumberOfTransitions | True |

| Group | Conformance Unit / Profile Title | Optional |
|-------------------------|--|----------|
| Aggregates | Aggregate Subscription – PercentBad | True |
| Aggregates | Aggregate Subscription – PercentGood | True |
| Aggregates | Aggregate Subscription – Range | True |
| Aggregates | Aggregate Subscription – Range2 | True |
| Aggregates | Aggregate Subscription – StandardDeviationPopulation | True |
| Aggregates | Aggregate Subscription – StandardDeviationSample | True |
| Aggregates | Aggregate Subscription – Start | True |
| Aggregates | Aggregate Subscription – StartBound | True |
| Aggregates | Aggregate Subscription – TimeAverage | True |
| Aggregates | Aggregate Subscription – TimeAverage2 | True |
| Aggregates | Aggregate Subscription – Total | True |
| Aggregates | Aggregate Subscription – Total2 | True |
| Aggregates | Aggregate Subscription – VariancePopulation | True |
| Aggregates | Aggregate Subscription – VarianceSample | True |
| Aggregates | Aggregate Subscription – WorstQuality | True |
| Aggregates | Aggregate Subscription – WorstQuality2 | True |
| Monitored Item Services | Monitor Aggregate Filter | False |

6.5.52 Nano Embedded Device Server Profile

Table 74 describes the details of the Nano Embedded Device *Server Profile*. This *Profile* is a *FullFeatured Profile* intended for chip level devices with limited resources. This *Profile* is functionally equivalent to the Core *Server Facet* and defines the OPC UA TCP binary protocol as the required transport profile.

Exposing types in the AddressSpace is optional for this Profile except if custom types (i.e. types that are derived from well-known ObjectTypes, VariableTypes, ReferenceType or DataTypes) are used. Exposing all supported types in the AddressSpace is mandatory in some higher level Profiles.

Table 74 – Nano Embedded Device Server Profile

| Group | Conformance Unit / Profile Title | Optional |
|------------------|----------------------------------|----------|
| Profile | Core Server Facet | False |
| Profile | UA-TCP UA-SC UA Binary | False |
| Base Information | Base Info Diagnostics | True |
| Base Information | Base Info Custom Type System | True |

6.5.53 Micro Embedded Device Server Profile

Table 75 describes the details of the Micro Embedded Device *Server Profile*. This *Profile* is a *FullFeatured Profile* intended for small devices with limited resources. This *Profile* builds upon the Nano Embedded Device *Server Profile*. The most important additions are: support for subscriptions via the Embedded Data Change *Subscription Server Facet* and support for at least two sessions. A complete Type System is not required; however, if the *Server* implements any non-UA types then these types and their super-types must be exposed.

Table 75 – Micro Embedded Device Server Profile

| Group | Conformance Unit / Profile Title | Optional |
|------------------|---|----------|
| Profile | Embedded DataChange Subscription Server Facet | False |
| Profile | Nano Embedded Device Server Profile | False |
| Session Services | Session Minimum 2 Parallel | False |

6.5.54 Embedded UA Server Profile

Table 76 describes the details of the Embedded UA *Server Profile*. This *Profile* is a *FullFeatured Profile* that is intended for devices with more than 50 MBs of memory and a more powerful processor. This *Profile* builds upon the Micro Embedded Device *Server Profile*. The most important additions are: support for security via the Security Policy – Basic128Rsa15 Facet, and support for the Standard DataChange *Subscription Server Facet*. This *Profile* also requires that servers expose all OPC-UA types that are used by the *Server* including their components and their super-types.

Table 76 – Embedded UA Server Profile

| Group | Conformance Unit / <i>Profile</i> Title | Optional |
|------------------|---|----------|
| <i>Profile</i> | Micro Embedded Device Server Profile | False |
| <i>Profile</i> | SecurityPolicy – Basic128Rsa15 | False |
| <i>Profile</i> | Standard DataChange Subscription Server Facet | False |
| <i>Profile</i> | User Token – X509 Certificate Server Facet | False |
| Base Information | Base Info Engineering Units | True |
| Base Information | Base Info Placeholder Modelling Rules | True |
| Base Information | Base Info Type System | False |
| Security | Security Default ApplicationInstanceCertificate | False |

6.5.55 Standard UA Server Profile

Table 77 describes the details of the Standard UA *Server Profile*. This *Profile* is a *FullFeatured Profile* that defines a minimum set of functionality required for PC based OPC UA servers. Such a server must provide the base *AddressSpace* structure with type nodes, instance nodes and diagnostic information. The *Server* must provide connection establishment through the OPC UA TCP binary protocol with security and the creation of at least 50 parallel sessions. It includes view services like browsing and the attribute services for reading and writing of current values. In addition, the monitoring of data changes is included with a minimum of 5 subscriptions for half of the required sessions (total 225) and a minimum of 500 monitored items for half of the subscriptions (total 56250).

Table 77 – Standard UA Server Profile

| Group | Conformance Unit / <i>Profile</i> Title | Optional |
|--------------------|---|----------|
| <i>Profile</i> | Embedded UA Server Profile | False |
| <i>Profile</i> | Enhanced DataChange Subscription Server Facet | False |
| Attribute Services | Attribute Write StatusCode & Timestamp | True |
| Base Information | Base Info Diagnostics | False |
| Discovery Services | Discovery Register | False |
| Discovery Services | Discovery Register2 | True |
| Session Services | Session Cancel | False |
| Session Services | Session Minimum 50 Parallel | False |
| View Services | View Minimum Continuation Point 05 | False |
| Session Services | <i>Session Change User</i> | True |

6.5.56 Global Discovery Server Profile

Table 78 describes the details of the Global Discovery *Server (GDS) Profile*. This *Profile* is a *FullFeatured Profile* that covers the necessary Services and Information Model of a UA Server that acts as a GDS.

Table 78 – Global Discovery Server Profile

| Group | Conformance Unit / <i>Profile</i> Title | Optional |
|----------------|---|----------|
| <i>Profile</i> | Core Server Facet | False |
| <i>Profile</i> | UA-TCP UA-SC UA Binary | False |
| <i>Profile</i> | SecurityPolicy – Basic128Rsa15 | False |

| Group | Conformance Unit / Profile Title | Optional |
|------------------|---|----------|
| Profile | SecurityPolicy – Basic256 | False |
| Profile | Standard DataChange Subscription Server Facet | False |
| Profile | User Token – X509 Certificate Server Facet | False |
| Profile | Method Server Facet | False |
| Security | Security Default ApplicationInstanceCertificate | False |
| Session Services | Session Minimum 50 Parallel | False |
| GDS | GDS Application Directory | False |
| GDS | GDS LDS-ME Connectivity | False |

6.5.57 Global Discovery and Certificate Management Server Profile

Table 79 describes the details of the Global Discovery and Certificate Management Server Profile. This Profile is a *FullFeatured Profile* that covers the necessary Services and Information Model of a UA Server that acts as a global Certificate Manager. The Certificate Manager can but does not have to be integrated with the GDS.

Table 79 – Global Discovery and Certificate Management Server Profile

| Group | Conformance Unit / Profile Title | Optional |
|---------|--|----------|
| Profile | Global Discovery Server Profile | False |
| Profile | SecurityPolicy – Basic256Sha256 | False |
| Profile | Standard Event Subscription Server Facet | False |
| Profile | Auditing Server Facet | False |
| Profile | File Access Server Facet | False |
| GDS | GDS Certificate Manager Pull Model | False |
| GDS | GDS Certificate Manager Push Model | False |

6.5.58 Core Client Facet

Table 80 describes the details of the Core Client Facet. This Facet defines the core functionality required for any Client. This Facet includes the core functions for Security and Session handling.

Table 80 – Core Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|------------------|--|----------|
| Profile | SecurityPolicy – Basic128Rsa15 | False |
| Profile | SecurityPolicy – None | False |
| Profile | User Token – User Name Password Client Facet | False |
| Profile | User Token – X509 Certificate Client Facet | False |
| Security | Security Administration | False |
| Session Services | Session Client Base | False |
| Session Services | Session Client Cancel | True |
| Session Services | Session Client Detect Shutdown | False |
| Session Services | Session Client General Service Behaviour | False |
| Session Services | Session Client Impersonate | True |
| Session Services | Session Client KeepAlive | False |
| Session Services | Session Client Renew NodeIds | True |

6.5.59 Request State Change Client Facet

Table 81 describes the details of the Request State Change Client Facet. This Facet specifies the ability to invoke the RequestServerStateChange Method.

Table 81 – Request State Change Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|------------------|---|----------|
| Base Information | Base Info Client RequestServerStateChange | False |

6.5.60 Global Certificate Management Client Facet

Table 82 describes the details of the Global Certificate Management *Client* Facet. This Facet defines the capability to interact with a *Global Certificate Management Server* to obtain an initial or renewed *Certificate* and *Trust Lists*.

Table 82 – Global Certificate Management Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|----------|--|----------|
| Security | Pull Model for Global Certificate and TrustList Management | True |
| Security | Push Model for Global Certificate and TrustList Management | True |
| Security | Pull or Push Model | False |

6.5.61 Base Client Behaviour Facet

Table 83 describes the details of the Base *Client* Behaviour Facet. This Facet indicates that the *Client* supports behaviour that *Clients* shall follow for best use by operators and administrators. They include allowing configuration of an endpoint for a server without using the discovery service set; Support for manual security setting configuration and behaviour with regard to security issues; support for Automatic reconnection to a disconnected server. These behaviours can only be tested in a test lab. They are best practice guidelines.

Table 83 – Base Client Behaviour Facet

| Group | Conformance Unit / Profile Title | Optional |
|-----------------------|--|----------|
| Discovery Services | Discovery Client Configure Endpoint | False |
| Security | Security Administration | False |
| Security | Security Administration – XML Schema | False |
| Security | Security Certificate Administration | False |
| Session Services | Session Client Auto Reconnect | True |
| Subscription Services | Subscription Client Multiple | False |
| Subscription Services | Subscription Client Publish Configurable | False |

6.5.62 Discovery Client Facet

Table 84 describes the details of the *Discovery Client* Facet. This Facet defines the ability to discover *Servers* and their Endpoints.

Table 84 – Discovery Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|--------------------|--|----------|
| Discovery Services | Discovery Client Configure Endpoint | False |
| Discovery Services | Discovery Client Find Servers Basic | False |
| Discovery Services | Discovery Client Find Servers Dynamic | False |
| Discovery Services | Discovery Client Find Servers with URI | True |
| | | |
| Discovery Services | Discovery Client Get Endpoints Basic | False |
| Discovery Services | Discovery Client Get Endpoints Dynamic | False |

6.5.63 Subnet Discovery Client Facet

Table 85 describes the details of the Subnet Discovery *Client* Facet. Support of this Facet enables discovery of the *Server* on a subnet.

Table 85 – Subnet Discovery Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|--------------------|---|----------|
| Discovery Services | Discovery Client Find Servers on Network using LDS-ME | True |
| Discovery Services | Discovery Client Find Servers on Network using mDNS | True |
| Discovery Services | Discovery Client Find Servers on Network | False |

6.5.64 Global Discovery Client Facet

Table 86 describes the details of the Global Discovery *Client* Facet. Support of this Facet enables system-wide discovery of *Servers* using a Global Discovery Server (GDS).

Table 86 – Global Discovery Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|--------------------|--------------------------------------|----------|
| Discovery Services | Discovery Client Find Servers in GDS | False |

6.5.65 AddressSpace Lookup Client Facet

Table 87 describes the details of the *AddressSpace* Lookup *Client* Facet. This Facet defines the ability to navigate through the *AddressSpace* and includes basic *AddressSpace* concepts, view and browse functionality and simple attribute read functionality.

Table 87 – AddressSpace Lookup Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|---------------------|--|----------|
| Address Space Model | Address Space Client Base | False |
| Attribute Services | Attribute Client Read Base | False |
| Attribute Services | Attribute Client Remote Nodes Attribute Access | True |
| Base Information | Base Info Client Basic | False |
| Base Information | Base Info Client GetMonitoredItems Method | True |
| View Services | View Client Basic Browse | False |
| View Services | View Client Basic ResultSet Filtering | False |
| View Services | View Client RegisterNodes | True |
| View Services | View Client TranslateBrowsePath | True |
| View Services | View Client Remote Nodes Browse | True |
| View Services | View Client Remote Nodes Translate Browse | True |

6.5.66 Entry-Level Support Client Facet

Note: This facet has been deprecated with the OPC UA Specification Version 1.03. It has been replaced by the Entry Level Support 2015 Client Facet.

6.5.67 Entry Level Support 2015 Client Facet

Table 88 describes the details of the Entry-Level Support *Client* Facet. This Facet defines the ability to interoperate with low-end *Servers*, in particular *Servers* that support the Nano Embedded Profile but in general *Servers* with defined limits.

Table 88 – Entry Level Support 2015 Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|-----------------------|--|----------|
| Base Information | Base Info Honour Server Operation Limits | False |
| Base Information | Base Info Client Type Pre-Knowledge | False |
| Session Services | Session Client Single Session | False |
| Subscription Services | Subscription Client Fallback | False |

6.5.68 Multi-Server Client Connection Facet

Table 89 describes the details of the Multi-Server *Client* Connection Facet. This Facet defines the ability for simultaneous access to multiple *Servers*.

Table 89 – Multi-Server Client Connection Facet

| Group | Conformance Unit / Profile Title | Optional |
|------------------|-------------------------------------|----------|
| Session Services | Session Client Multiple Connections | False |

6.5.69 File Access Client Facet

Table 90 describes the details of the File Access *Client* Facet. This Facet defines the ability to use File transfer via the defined FileType. This includes reading and optionally writing.

Table 90 – File Access Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|------------------|----------------------------------|----------|
| Base Information | Base Info Client FileType Base | False |
| Base Information | Base Info Client FileType Write | True |

6.5.70 Documentation – Client

Table 91 describes the details of the Documentation – *Client*. This Facet provides a list of user documentation that a *Client* application should provide.

Table 91 – Documentation – Client

| Group | Conformance Unit / Profile Title | Optional |
|---------------|---|----------|
| Miscellaneous | Documentation Client – Installation | False |
| Miscellaneous | Documentation Client – Multiple Languages | True |
| Miscellaneous | Documentation Client – On-line | True |
| Miscellaneous | Documentation Client – Supported Profiles | True |
| Miscellaneous | Documentation Client – Trouble Shooting Guide | True |
| Miscellaneous | Documentation Client – Users Guide | False |

6.5.71 Attribute Read Client Facet

Table 92 describes the details of the *Attribute* Read *Client* Facet. This Facet defines the ability to read *Attribute* values of *Nodes*.

Table 92 – Attribute Read Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|--------------------|--|----------|
| Attribute Services | Attribute Client Read Base | False |
| Attribute Services | Attribute Client Read Complex | True |
| Attribute Services | Attribute Client Read with proper Encoding | True |

6.5.72 Attribute Write Client Facet

Table 93 describes the details of the *Attribute* Write *Client* Facet. This Facet defines the ability to write *Attribute* values of *Nodes*.

Table 93 – Attribute Write Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|--------------------|--|----------|
| Attribute Services | Attribute Client Write Base | False |
| Attribute Services | Attribute Client Write Complex | True |
| Attribute Services | Attribute Client Write Quality & Timestamp | True |

6.5.73 DataChange Subscriber Client Facet

Table 94 describes the details of the DataChange Subscriber *Client* Facet. This Facet defines the ability to monitor *Attribute* values for data change.

Table 94 – DataChange Subscriber Client Facet

| Group | Conformance Unit / <i>Profile</i> Title | Optional |
|-------------------------|---|----------|
| Monitored Item Services | Monitor Client by Index | False |
| Monitored Item Services | Monitor Client Deadband Filter | True |
| Monitored Item Services | Monitor Client Modify | True |
| Monitored Item Services | Monitor Client Trigger | True |
| Monitored Item Services | Monitor Client Value Change | False |
| Subscription Services | Subscription Client Basic | False |
| Subscription Services | Subscription Client Modify | True |
| Subscription Services | Subscription Client Multiple | True |
| Subscription Services | Subscription Client Republish | False |

6.5.74 Durable Subscription Client Facet

Table 95 describes the details of the Durable *Subscription Client* Facet. This Facet specifies use of durable *Subscriptions*. It implies support of any of the DataChange or Event Subscriber Facets.

Table 95 – Durable Subscription Client Facet

| Group | Conformance Unit / <i>Profile</i> Title | Optional |
|-----------------------|---|----------|
| Subscription Services | Subscription Client Durable | False |

6.5.75 DataAccess Client Facet

Table 96 describes the details of the DataAccess *Client* Facet. This Facet defines the ability to utilize the DataAccess Information Model, i.e., industrial automation data like analog and discrete data items and their quality of service.

Table 96 – DataAccess Client Facet

| Group | Conformance Unit / <i>Profile</i> Title | Optional |
|---------------------|--|----------|
| Address Space Model | Address Space Client Base | False |
| Address Space Model | Address Space Client Complex DataTypes | True |
| Attribute Services | Attribute Client Read Base | False |
| Attribute Services | Attribute Client Read Complex | True |
| Attribute Services | Attribute Client Read with proper Encoding | True |
| Data Access | Data Access Client Basic | False |
| Data Access | Data Access Client Deadband | True |
| Data Access | Data Access Client SemanticChange | True |

6.5.76 Event Subscriber Client Facet

Table 97 describes the details of the *Event* Subscriber *Client* Facet. This Facet defines the ability to subscribe for *Event Notifications*. This includes basic *AddressSpace* concept and the browsing of it, adding events and event filters as monitored items and adding subscriptions.

Table 97 – Event Subscriber Client Facet

| Group | Conformance Unit / <i>Profile</i> Title | Optional |
|-------------------------|---|----------|
| Address Space Model | Address Space Client Base | False |
| Monitored Item Services | Monitor Client Complex Event Filter | True |
| Monitored Item Services | Monitor Client Event Filter | False |
| Monitored Item Services | Monitor Client Events | False |
| Monitored Item Services | Monitor Client Modify | True |
| Monitored Item Services | Monitor Client Trigger | True |

| Group | Conformance Unit / Profile Title | Optional |
|-----------------------|----------------------------------|----------|
| Subscription Services | Subscription Client Basic | False |
| Subscription Services | Subscription Client Modify | True |
| Subscription Services | Subscription Client Multiple | True |
| Subscription Services | Subscription Client Republish | False |
| View Services | View Client Basic Browse | True |
| View Services | View Client TranslateBrowsePath | True |

6.5.77 Base Event Processing Client Facet

Table 98 describes the details of the Base *Event Processing Client* Facet. This Facet defines the ability to subscribe for and process basic OPC UA *Events*. The Client has to support at least one of the *Events* in the Facet.

Table 98 – Base Event Processing Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|------------------|--|----------|
| <i>Profile</i> | Event Subscriber Client Facet | False |
| Base Information | Base Info Client System Status | True |
| Base Information | Base Info Client System Status Underlying System | True |
| Base Information | Base Info Client Device Failure | True |
| Base Information | Base Info Client Progress Events | True |
| Base Information | Base Info Client Change Events | True |
| Base Information | Base Info Event Processing | False |

6.5.78 Notifier and Source Hierarchy Client Facet

Table 99 describes the details of the Notifier and Source Hierarchy *Client* Facet. This Facet defines the ability to find and use a hierarchy of *Objects* that are event notifier and *Nodes* that are event sources in the *Server AddressSpace*.

Table 99 – Notifier and Source Hierarchy Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|-----------------------|--|----------|
| <i>Profile</i> | Event Subscriber Client Facet | False |
| Address Space Model | Address Space Client Notifier Hierarchy | False |
| Address Space Model | Address Space Client Source Hierarchy | False |
| Subscription Services | Subscription Client Publish Configurable | False |

6.5.79 A & C Base Condition Client Facet

Table 100 describes the details of the A & C Base Condition Client Facet. This Facet defines the ability to use the *Alarm* and *Condition* basic model. This includes the ability to subscribe for Events and to initiate a Refresh *Method*.

Table 100 – A & C Base Condition Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|-----------------------|----------------------------------|----------|
| <i>Profile</i> | Event Subscriber Client Facet | False |
| <i>Profile</i> | Method Client Facet | False |
| Alarms and Conditions | A & C Basic Client | False |
| Alarms and Conditions | A & C ConditionClasses Client | False |
| Alarms and Conditions | A & C Refresh Client | False |

6.5.80 A & C Refresh2 Client Facet

Table 101 describes the details of the A & C Refresh2 Client Facet. This Facet enhances the A & C Base Condition Server Facet with the ability to initiate a ConditionRefresh2 *Method*.

Table 101 – A & C Reresh2 Client Facet

| Group | Conformance Unit / <i>Profile</i> Title | Optional |
|-----------------------|---|----------|
| <i>Profile</i> | A & C Base Condition Client Facet | False |
| Alarms and Conditions | A & C Refresh2 Client | False |

6.5.81 A & C Address Space Instance Client Facet

Table 102 describes the details of the A & C Address Space Instance *Client* Facet. This Facet defines the ability to use *Condition* instances in the *AddressSpace*.

Table 102 – A & C Address Space Instance Client Facet

| Group | Conformance Unit / <i>Profile</i> Title | Optional |
|-----------------------|---|----------|
| Alarms and Conditions | A & C Instances Client | False |

6.5.82 A & C Enable Client Facet

Table 103 describes the details of the A & C Enable *Client* Facet. This Facet defines the ability to enable and disable *Alarms*,

Table 103 – A & C Enable Client Facet

| Group | Conformance Unit / <i>Profile</i> Title | Optional |
|-----------------------|---|----------|
| <i>Profile</i> | A & C Base Condition Client Facet | False |
| Alarms and Conditions | A & C Enable Client | False |

6.5.83 A & C Alarm Client Facet

Table 104 describes the details of the A & C *Alarm Client* Facet. This Facet defines the ability to use the alarming model (the *AlarmType* or any of the sub-types).

Table 104 – A & C Alarm Client Facet

| Group | Conformance Unit / <i>Profile</i> Title | Optional |
|-----------------------|---|----------|
| <i>Profile</i> | A & C Base Condition Client Facet | False |
| Alarms and Conditions | A & C Acknowledge Client | False |
| Alarms and Conditions | A & C Alarm Client | False |
| Alarms and Conditions | A & C Comment Client | True |
| Alarms and Conditions | A & C Confirm Client | True |
| Alarms and Conditions | A & C Discrete Client | False |
| Alarms and Conditions | A & C OffNormal Client | True |
| Alarms and Conditions | A & C SystemOffNormal Client | True |
| Alarms and Conditions | A & C Shelving Client | True |
| Alarms and Conditions | A & C Trip Client | True |

6.5.84 A & C Exclusive Alarming Client Facet

Table 105 describes the details of the A & C Exclusive Alarming *Client* Facet. This Facet defines the ability to use the exclusive *Alarm* model. This includes understanding the various subtypes such as *ExclusiveRateOfChangeAlarm*, *ExclusiveLevelAlarm* and *ExclusiveDeviationAlarm*.

Table 105 – A & C Exclusive Alarming Client Facet

| Group | Conformance Unit / <i>Profile</i> Title | Optional |
|-----------------------|---|----------|
| <i>Profile</i> | A & C Alarm Client Facet | False |
| Alarms and Conditions | A & C Exclusive Deviation Client | True |
| Alarms and Conditions | A & C Exclusive Level Client | True |
| Alarms and Conditions | A & C Exclusive Limit Client | False |
| Alarms and Conditions | A & C Exclusive RateOfChange Client | True |

6.5.85 A & C Non-Exclusive Alarming Client Facet

Table 106 describes the details of the A & C Non-Exclusive Alarming *Client* Facet. This Facet defines the ability to use the non-exclusive *Alarm* model. This includes understanding the various subtypes such as *NonExclusiveRateOfChangeAlarm*, *NonExclusiveLevelAlarm* and *NonExclusiveDeviationAlarm*.

Table 106 – A & C Non-Exclusive Alarming Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|-----------------------|---|----------|
| <i>Profile</i> | A & C Alarm Client Facet | False |
| Alarms and Conditions | A & C Non-Exclusive Deviation Client | True |
| Alarms and Conditions | A & C Non-Exclusive Level Client | True |
| Alarms and Conditions | A & C Non-Exclusive Limit Client | False |
| Alarms and Conditions | A & C Non-Exclusive RateOfChange Client | True |

6.5.86 A & C Previous Instances Client Facet

Table 107 describes the details of the A & C Previous Instances *Client* Facet. This Facet defines the ability to use previous instances of *Alarms*. This implies the ability to understand *branchIds*.

Table 107 – A & C Previous Instances Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|-----------------------|-----------------------------------|----------|
| <i>Profile</i> | A & C Base Condition Client Facet | False |
| Alarms and Conditions | A & C Branch Client | False |

6.5.87 A & C Dialog Client Facet

Table 108 describes the details of the A & C Dialog *Client* Facet. This Facet defines the ability to use the dialog model. This implies the support of *Method* invocation to respond to dialog messages.

Table 108 – A & C Dialog Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|-----------------------|-----------------------------------|----------|
| <i>Profile</i> | A & C Base Condition Client Facet | False |
| Alarms and Conditions | A & C Dialog Client | False |

6.5.88 A & C CertificateExpiration Client Facet

Table 109 describes the details of the A & C CertificateExpiration *Client* Facet. This Facet defines the ability to use the *CertificateExpirationAlarmType*.

Table 109 – A & C CertificateExpiration Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|-----------------------|------------------------------------|----------|
| <i>Profile</i> | A & C Base Condition Client Facet | False |
| Alarms and Conditions | A & C Alarm Client | False |
| Alarms and Conditions | A & C Acknowledge Client | True |
| Alarms and Conditions | A & C Comment Client | True |
| Alarms and Conditions | A & C Confirm Client | True |
| Alarms and Conditions | A & C Shelving Client | True |
| Alarms and Conditions | A & C CertificateExpiration Client | False |

6.5.89 A & E Proxy Facet

Table 110 describes the details of the A & E Proxy Facet. This Facet describes the functionality used by a default A & E *Client* proxy. A *Client* exposes this Facet so that a *Server* may be able to better understand the commands that are being issued by the *Client*, since this Facet indicates that the *Client* is an A&E Com *Client*.

Table 110 – A & E Proxy Facet

| Group | Conformance Unit / Profile Title | Optional |
|-------------------------|--|-----------------|
| Address Space Model | Address Space Client Base | False |
| Alarms and Conditions | A & C Acknowledge Client | False |
| Alarms and Conditions | A & C Alarm Client | False |
| Alarms and Conditions | A & C Basic Client | False |
| Alarms and Conditions | A & C ConditionClasses Client | False |
| Alarms and Conditions | A & C Discrete Client | False |
| Alarms and Conditions | A & C Exclusive Deviation Client | False |
| Alarms and Conditions | A & C Exclusive Level Client | False |
| Alarms and Conditions | A & C Exclusive Limit Client | False |
| Alarms and Conditions | A & C Exclusive RateOfChange Client | False |
| Alarms and Conditions | A & C Instances Client | False |
| Alarms and Conditions | A & C Non-Exclusive Deviation Client | False |
| Alarms and Conditions | A & C Non-Exclusive Level Client | False |
| Alarms and Conditions | A & C Non-Exclusive Limit Client | False |
| Alarms and Conditions | A & C Non-Exclusive RateOfChange Client | False |
| Alarms and Conditions | A & C OffNormal Client | False |
| Alarms and Conditions | A & C SystemOffNormal Client | True |
| Alarms and Conditions | A & C Refresh Client | False |
| Alarms and Conditions | A & C Trip Client | False |
| Attribute Services | Attribute Client Read Base | False |
| Base Information | Base Info Client Basic | False |
| Base Information | Base Info Client Change Events | False |
| Discovery Services | Discovery Client Configure Endpoint | False |
| Discovery Services | Discovery Client Find Servers Basic | False |
| Discovery Services | Discovery Client Find Servers Dynamic | False |
| Discovery Services | Discovery Client Find Servers with URI | False |
| Discovery Services | Discovery Client Get Endpoints Basic | False |
| Discovery Services | Discovery Client Get Endpoints Dynamic | False |
| Method Services | Method Client Call | False |
| Monitored Item Services | Monitor Client Complex Event Filter | False |
| Monitored Item Services | Monitor Client Event Filter | False |
| Monitored Item Services | Monitor Client Events | False |
| Security | Security Administration | False |
| Security | Security Administration – XML Schema | False |
| Security | Security Certificate Administration | False |
| Session Services | Session Client Auto Reconnect | False |
| Subscription Services | Subscription Client Basic | False |
| Subscription Services | Subscription Client Multiple | False |
| Subscription Services | Subscription Client Publish Configurable | False |
| Subscription Services | Subscription Client Republish | False |
| View Services | View Client Basic Browse | False |
| View Services | View Client Basic ResultSet Filtering | False |
| View Services | View Client TranslateBrowsePath | False |

6.5.90 Method Client Facet

Table 111 describes the details of the *Method Client* Facet. This Facet defines the ability to call arbitrary *Methods*.

Table 111 – Method Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|-----------------|---|-----------------|
| Method Services | Method Client Call | False |

6.5.91 Auditing Client Facet

Table 112 describes the details of the Auditing *Client* Facet. This Facet defines the ability to monitor *AuditEvents*.

Table 112 – Auditing Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|----------|----------------------------------|----------|
| Profile | Event Subscriber Client Facet | False |
| Auditing | Auditing Client Audit ID | False |
| Auditing | Auditing Client Subscribes | False |

6.5.92 Node Management Client Facet

Table 113 describes the details of the Node Management *Client* Facet. This Facet defines the ability to configure the *AddressSpace* of an OPC UA *Server* through OPC UA *Node* Management *Service* Set.

Table 113 – Node Management Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|--------------------------|----------------------------------|----------|
| Address Space Model | Address Space Client Base | False |
| Node Management Services | Node Management Client | False |

6.5.93 Advanced Type Programming Client Facet

Table 114 describes the details of the Advanced Type Programming *Client* Facet. This Facet defines the ability to use the type model and process the instance *AddressSpace* based on the type model. For example a client may contain generic displays that are based on a type, in that they contain a relative path from some main type. On call up this main type is matched to an instance and all of display items are resolved based on the provided type model.

Table 114 – Advanced Type Programming Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|---------------------|-----------------------------------|----------|
| Address Space Model | Address Space Client Base | False |
| Base Information | Base Info Client Basic | False |
| Base Information | Base Info Client Type Programming | False |
| View Services | View Client TranslateBrowsePath | False |

6.5.94 Diagnostic Client Facet

Table 115 describes the details of the Diagnostic *Client* Facet. This Facet defines the ability to read and process diagnostic information that is part of the OPC UA information model.

Table 115 – Diagnostic Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|---------------------|----------------------------------|----------|
| Address Space Model | Address Space Client Base | False |
| Base Information | Base Info Client Basic | False |
| Base Information | Base Info Client Diagnostics | False |

6.5.95 Redundant Client Facet

Table 116 describes the details of the Redundant *Client* Facet. This Facet defines the ability to use the redundancy feature available for redundant *Clients*.

Table 116 – Redundant Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|------------|----------------------------------|----------|
| Redundancy | Redundancy Client | False |

| Group | Conformance Unit / Profile Title | Optional |
|-----------------------|---|----------|
| Subscription Services | Subscription Client TransferSubscriptions | True |

6.5.96 Redundancy Switch Client Facet

Table 117 describes the details of the Redundancy Switch *Client* Facet. A *Client* that supports this Facet supports monitoring the redundancy status for non-transparent redundant *Servers* and switching to the backup *Server* when they recognize a change.

Table 117 – Redundancy Switch Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|------------|----------------------------------|----------|
| Redundancy | Redundancy Client Switch | False |

6.5.97 Historical Access Client Facet

Table 118 describes the details of the Historical Access *Client* Facet. This Facet defines the ability to read, process, and update historical data.

Table 118 – Historical Access Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|--------------------|-----------------------------------|----------|
| Attribute Services | Attribute Client Historical Read | False |
| Historical Access | Historical Access Client Browse | False |
| Historical Access | Historical Access Client Read Raw | False |

6.5.98 Historical Annotation Client Facet

Table 119 describes the details of the Historical Annotation *Client* Facet. This Facet defines the ability to retrieve and write annotations for historical data.

Table 119 – Historical Annotation Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|-------------------|--------------------------------------|----------|
| Profile | Historical Access Client Facet | False |
| Profile | Historical Data Update Client Facet | False |
| Historical Access | Historical Access Client Annotations | False |

6.5.99 Historical Data AtTime Client Facet

Table 120 describes the details of the Historical Data AtTime *Client* Facet. This Facet defines the ability to access data at specific instances in time.

Table 120 – Historical Data AtTime Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|-------------------|--|----------|
| Profile | Historical Access Client Facet | False |
| Historical Access | Historical Access Client Time Instance | False |

6.5.100 Historical Aggregate Client Facet

Table 121 describes the details of the Historical Aggregate *Client* Facet. This Facet defines the ability to read historical data by specifying the needed aggregate. This implies consideration of the list of aggregates supported by the *Server*.

Table 121 – Historical Aggregate Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|------------|--------------------------------------|----------|
| Aggregates | Aggregate – Client AnnotationCount | True |
| Aggregates | Aggregate – Client Average | True |
| Aggregates | Aggregate – Client Count | True |
| Aggregates | Aggregate – Client Custom Aggregates | True |

| Group | Conformance Unit / Profile Title | Optional |
|-------------------|--|----------|
| Aggregates | Aggregate – Client Delta | True |
| Aggregates | Aggregate – Client DeltaBounds | True |
| Aggregates | Aggregate – Client DurationBad | True |
| Aggregates | Aggregate – Client DurationGood | True |
| Aggregates | Aggregate – Client DurationInStateNonZero | True |
| Aggregates | Aggregate – Client DurationInStateZero | True |
| Aggregates | Aggregate – Client End | True |
| Aggregates | Aggregate – Client EndBound | True |
| Aggregates | Aggregate – Client Interpolative | True |
| Aggregates | Aggregate – Client Maximum | True |
| Aggregates | Aggregate – Client Maximum2 | True |
| Aggregates | Aggregate – Client MaximumActualTime | True |
| Aggregates | Aggregate – Client MaximumActualTime2 | True |
| Aggregates | Aggregate – Client Minimum | True |
| Aggregates | Aggregate – Client Minimum2 | True |
| Aggregates | Aggregate – Client MinimumActualTime | True |
| Aggregates | Aggregate – Client MinimumActualTime2 | True |
| Aggregates | Aggregate – Client NumberOfTransitions | True |
| Aggregates | Aggregate – Client PercentBad | True |
| Aggregates | Aggregate – Client PercentGood | True |
| Aggregates | Aggregate – Client Range | True |
| Aggregates | Aggregate – Client Range2 | True |
| Aggregates | Aggregate – Client StandardDeviationPopulation | True |
| Aggregates | Aggregate – Client StandardDeviationSample | True |
| Aggregates | Aggregate – Client Start | True |
| Aggregates | Aggregate – Client StartBound | True |
| Aggregates | Aggregate – Client TimeAverage | True |
| Aggregates | Aggregate – Client TimeAverage2 | True |
| Aggregates | Aggregate – Client Total | True |
| Aggregates | Aggregate – Client Total2 | True |
| Aggregates | Aggregate – Client Usage | False |
| Aggregates | Aggregate – Client VariancePopulation | True |
| Aggregates | Aggregate – Client VarianceSample | True |
| Aggregates | Aggregate – Client WorstQuality | True |
| Aggregates | Aggregate – Client WorstQuality2 | True |
| Historical Access | Historical Access Client Read Aggregates | False |

6.5.101 Historical Data Update Client Facet

Table 122 describes the details of the Historical Data Update *Client* Facet. This Facet defines the ability to update historical data.

Table 122 – Historical Data Update Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|--------------------|--------------------------------------|----------|
| Attribute Services | Attribute Client Historical Updates | False |
| Historical Access | Historical Access Client Data Update | False |

6.5.102 Historical Data Replace Client Facet

Table 122 describes the details of the Historical Data Replace *Client* Facet. This Facet defines the ability to replace historical data.

Table 123 – Historical Data Replace Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|--------------------|---------------------------------------|----------|
| Attribute Services | Attribute Client Historical Updates | False |
| Historical Access | Historical Access Client Data Replace | False |

6.5.103 Historical Data Insert Client Facet

Table 124 describes the details of the Historical Data Insert *Client* Facet. This Facet defines the ability to insert historical data.

Table 124 – Historical Data Insert Client Facet

| Group | Conformance Unit / <i>Profile</i> Title | Optional |
|--------------------|---|----------|
| Attribute Services | Attribute Client Historical Updates | False |
| Historical Access | Historical Access Client Data Insert | False |

6.5.104 Historical Data Delete Client Facet

Table 125 describes the details of the Historical Data Delete *Client* Facet. This Facet defines the ability to delete historical data.

Table 125 – Historical Data Delete Client Facet

| Group | Conformance Unit / <i>Profile</i> Title | Optional |
|--------------------|---|----------|
| Attribute Services | Attribute Client Historical Updates | False |
| Historical Access | Historical Access Client Data Delete | False |

6.5.105 Historical Access Client Server Timestamp Facet

Table 126 describes the details of the Historical Access *Client Server* Timestamp Facet. This Facet defines the ability to request and process *Server* timestamps, in addition to source timestamps.

Table 126 – Historical Access Client Server Timestamp Facet

| Group | Conformance Unit / <i>Profile</i> Title | Optional |
|-------------------|---|----------|
| Historical Access | Historical Access Client Server Timestamp | False |

6.5.106 Historical Access Modified Data Client Facet

Table 127 describes the details of the Historical Access Modified Data *Client* Facet. This Facet defines the ability to access prior historical data (values that were modified or inserted).

Table 127 – Historical Access Modified Data Client Facet

| Group | Conformance Unit / <i>Profile</i> Title | Optional |
|-------------------|---|----------|
| <i>Profile</i> | Historical Access Client Facet | False |
| Historical Access | Historical Access Client Read Modified | False |

6.5.107 Structured Data AtTime Client Facet

Table 128 describes the details of the Historical Structured Data AtTime *Client* Facet. This Facet defines the ability to read structured values for historical nodes at specific instances in time.

Table 128 – Historical Structured Data AtTime Client Facet

| Group | Conformance Unit / <i>Profile</i> Title | Optional |
|-------------------|---|----------|
| <i>Profile</i> | Historical Data AtTime Client Facet | False |
| Historical Access | Historical Access Client Structure Data Time Instance | False |

6.5.108 Historical Structured Data Access Client Facet

Table 129 describes the details of the Historical Structured Data Access *Client* Facet. This Facet defines the ability to read structured values for historical nodes.

Table 129 – Historical Structured Data Access Client Facet

| Group | Conformance Unit / <i>Profile</i> Title | Optional |
|-------------------|---|----------|
| <i>Profile</i> | Historical Access Client Facet | False |
| Historical Access | Historical Access Client Structure Data Raw | False |

6.5.109 Historical Structured Data Modified Client Facet

Table 130 describes the details of the Historical Structured Data Modified *Client* Facet. This Facet defines the ability to read structured values for prior historical data (values that were modified or inserted).

Table 130 – Historical Structured Data Modified Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|-------------------|---|----------|
| Profile | Historical Access Modified Data Client Facet | False |
| Historical Access | Historical Access Client Structure Data Read Modified | False |

6.5.110 Historical Structured Data Delete Client Facet

Table 131 describes the details of the Historical Structured Data Delete *Client* Facet. This Facet defines the ability to remove structured historical data.

Table 131 – Historical Structured Data Delete Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|-------------------|--|----------|
| Profile | Historical Data Delete Client Facet | False |
| Historical Access | Historical Access Client Structure Data Delete | False |

6.5.111 Historical Structured Data Update Client Facet

Table 132 describes the details of the Historical Structure Data Update *Client* Facet. This Facet defines the ability to update structured historical data.

Table 132 – Historical Structured Data Update Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|-------------------|--|----------|
| Profile | Historical Data Update Client Facet | False |
| Historical Access | Historical Access Client Structure Data Update | False |

6.5.112 Historical Structured Data Replace Client Facet

Table 132 describes the details of the Historical Structure Data Replace *Client* Facet. This Facet defines the ability to replace structured historical data.

Table 133 – Historical Structured Data Replace Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|-------------------|---|----------|
| Profile | Historical Data Update Client Facet | False |
| Historical Access | Historical Access Client Structure Data Replace | False |

6.5.113 Historical Structured Data Insert Client Facet

Table 134 describes the details of the Historical Structured Data Insert *Client* Facet. This Facet defines the ability to insert structured historical data.

Table 134 – Historical Structured Data Insert Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|-------------------|--|----------|
| Profile | Historical Data Insert Client Facet | False |
| Historical Access | Historical Access Client Structure Data Insert | False |

6.5.114 Historical Events Client Facet

Table 135 describes the details of the Historical Events *Client* Facet. This Facet defines the ability to read Historical Events, including simple filtering.

Table 135 – Historical Events Client Facet

| Group | Conformance Unit / <i>Profile</i> Title | Optional |
|--------------------|---|----------|
| Attribute Services | Attribute Client Historical Read | False |
| Historical Access | Historical Access Client Read Events | False |

6.5.115 Historical Event Update Client Facet

Table 136 describes the details of the Historical *Event* Update *Client* Facet. This Facet defines the ability to update historical events.

Table 136 – Historical Event Update Client Facet

| Group | Conformance Unit / <i>Profile</i> Title | Optional |
|--------------------|---|----------|
| Attribute Services | Attribute Client Historical Updates | False |
| Historical Access | Historical Access Client Event Updates | False |

6.5.116 Historical Event Replace Client Facet

Table 136 describes the details of the Historical *Event* Replace *Client* Facet. This Facet defines the ability to replace historical events.

Table 137 – Historical Event Replace Client Facet

| Group | Conformance Unit / <i>Profile</i> Title | Optional |
|--------------------|---|----------|
| Attribute Services | Attribute Client Historical Updates | False |
| Historical Access | Historical Access Client Event Replaces | False |

6.5.117 Historical Event Delete Client Facet

Table 138 describes the details of the Historical *Event* Delete *Client* Facet. This Facet defines the ability to delete Historical events.

Table 138 – Historical Event Delete Client Facet

| Group | Conformance Unit / <i>Profile</i> Title | Optional |
|--------------------|---|----------|
| Attribute Services | Attribute Client Historical Updates | False |
| Historical Access | Historical Access Client Event Deletes | False |

6.5.118 Historical Event Insert Client Facet

Table 139 describes the details of the Historical *Event* Insert *Client* Facet. This Facet defines the ability to insert historical events.

Table 139 – Historical Event Insert Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|--------------------|--|----------|
| Attribute Services | Attribute Client Historical Updates | False |
| Historical Access | Historical Access Client Event Inserts | False |

6.5.119 Aggregate Subscriber Client Facet

Table 140 describes the details of the Aggregate Subscriber *Client* Facet. This Facet defines the ability to use the aggregate filter when subscribing for *Attribute* values.

Table 140 – Aggregate Subscriber Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|------------|---|----------|
| Aggregates | Aggregate Subscription – Client DeltaBounds | True |
| Aggregates | Aggregate Subscription – Client AnnotationCount | True |
| Aggregates | Aggregate Subscription – Client Average | True |
| Aggregates | Aggregate Subscription – Client Count | True |
| Aggregates | Aggregate Subscription – Client Custom Aggregates | True |
| Aggregates | Aggregate Subscription – Client Delta | True |
| Aggregates | Aggregate Subscription – Client DurationBad | True |
| Aggregates | Aggregate Subscription – Client DurationGood | True |
| Aggregates | Aggregate Subscription – Client DurationInStateNonZero | True |
| Aggregates | Aggregate Subscription – Client DurationInStateZero | True |
| Aggregates | Aggregate Subscription – Client End | True |
| Aggregates | Aggregate Subscription – Client EndBound | True |
| Aggregates | Aggregate Subscription – Client Filter | False |
| Aggregates | Aggregate Subscription – Client Interpolative | True |
| Aggregates | Aggregate Subscription – Client Maximum | True |
| Aggregates | Aggregate Subscription – Client Maximum2 | True |
| Aggregates | Aggregate Subscription – Client MaximumActualTime | True |
| Aggregates | Aggregate Subscription – Client MaximumActualTime2 | True |
| Aggregates | Aggregate Subscription – Client Minimum | True |
| Aggregates | Aggregate Subscription – Client Minimum2 | True |
| Aggregates | Aggregate Subscription – Client MinimumActualTime | True |
| Aggregates | Aggregate Subscription – Client MinimumActualTime2 | True |
| Aggregates | Aggregate Subscription – Client NumberOfTransitions | True |
| Aggregates | Aggregate Subscription – Client PercentBad | True |
| Aggregates | Aggregate Subscription – Client PercentGood | True |
| Aggregates | Aggregate Subscription – Client Range | True |
| Aggregates | Aggregate Subscription – Client Range2 | True |
| Aggregates | Aggregate Subscription – Client StandardDeviationPopulation | True |
| Aggregates | Aggregate Subscription – Client StandardDeviationSample | True |
| Aggregates | Aggregate Subscription – Client Start | True |
| Aggregates | Aggregate Subscription – Client StartBound | True |
| Aggregates | Aggregate Subscription – Client TimeAverage | True |
| Aggregates | Aggregate Subscription – Client TimeAverage2 | True |
| Aggregates | Aggregate Subscription – Client Total | True |
| Aggregates | Aggregate Subscription – Client Total2 | True |

| Group | Conformance Unit / Profile Title | Optional |
|-------------------------|--|----------|
| Aggregates | Aggregate Subscription – Client VariancePopulation | True |
| Aggregates | Aggregate Subscription – Client VarianceSample | True |
| Aggregates | Aggregate Subscription – Client WorstQuality | True |
| Aggregates | Aggregate Subscription – Client WorstQuality2 | True |
| Monitored Item Services | Monitor Client Aggregate Filter | False |
| Monitored Item Services | Monitor Client by Index | False |
| Monitored Item Services | Monitor Client Modify | True |
| Monitored Item Services | Monitor Client Value Change | False |
| Subscription Services | Subscription Client Basic | False |
| Subscription Services | Subscription Client Modify | True |
| Subscription Services | Subscription Client Multiple | True |
| Subscription Services | Subscription Client Republish | True |

6.5.120 Global Certificate Management Client Profile

Table 141 describes the details of the Global Certificate Management *Client Profile*. This *Profile* is a *FullFeatured Profile* that uses the Push Model for the management of *Certificates* and *Trust Lists*.

Table 141 – Global Certificate Management Client Profile

| Group | Conformance Unit / Profile Title | Optional |
|----------------|---|----------|
| <i>Profile</i> | Core Client Facet | False |
| <i>Profile</i> | UA-TCP UA-SC UA Binary | False |
| <i>Profile</i> | SecurityPolicy – Basic256 | False |
| <i>Profile</i> | SecurityPolicy – Basic256Sha256 | False |
| <i>Profile</i> | Discovery Client Facet | False |
| <i>Profile</i> | Entry-Level Support Client Facet | False |
| <i>Profile</i> | Method Client Facet | False |
| <i>Profile</i> | File Access Client Facet | False |
| Security | Security Default ApplicationInstanceCertificate | False |
| GDS | Certificate Manager Push Model | False |

6.5.121 Standard UA Client Profile

Table 142 describes the details of the Standard UA *Client Profile*. This *Profile* is a *FullFeatured Profile* that defines a minimum set of functionality required for generic OPC UA *Clients*. Such a *Client* shall be able to use local, subnet and global discovery. It shall be able to maintain a connection with a single *Session* (as required for nano embedded *Servers*). If *Subscriptions* are used, the *Client* shall respect the limits of *Servers* with limited resources. If a *Server* does not support *Subscriptions*, the *Client* shall provide read access as fallback. The *Client* must provide connection establishment through the OPC UA TCP binary protocol with and without security.

Table 142 – Standard UA Client Profile

| Group | Conformance Unit / Profile Title | Optional |
|----------------|--|----------|
| <i>Profile</i> | Core Client Facet | False |
| <i>Profile</i> | Base Client Behaviour Facet | False |
| <i>Profile</i> | Discovery Client Facet | False |
| <i>Profile</i> | Subnet Discovery Client Facet | False |
| <i>Profile</i> | Global Discovery Client Facet | False |
| <i>Profile</i> | Global Certificate Management Client Facet | False |
| <i>Profile</i> | AddressSpace Lookup Client Facet | False |
| <i>Profile</i> | Entry Level Support 2015 Client Facet | False |
| <i>Profile</i> | Attribute Read Client Facet | False |
| <i>Profile</i> | Attribute Write Client Facet | False |
| <i>Profile</i> | Method Client Facet | False |
| <i>Profile</i> | DataChange Subscriber Client Facet | False |

| Group | Conformance Unit / Profile Title | Optional |
|---------|----------------------------------|----------|
| Profile | SecurityPolicy = Basic256 | False |
| Profile | SecurityPolicy = Basic256Sha256 | False |
| Profile | UA-TCP UA-SC UA Binary | False |
| Profile | User Token – Anonymous Facet | False |

6.5.122 User Token – Anonymous Facet

Table 143 describes the details of the User Token – Anonymous Facet. This Facet indicates that anonymous User Tokens are supported.

Table 143 – User Token – Anonymous Facet

| Group | Conformance Unit / Profile Title | Optional |
|----------|----------------------------------|----------|
| Security | Security User Anonymous | False |

6.5.123 User Token – User Name Password Server Facet

Table 144 describes the details of the User Token – User Name Password *Server* Facet. This Facet indicates that a user token that is comprised of a username and password is supported. This User Token can affect the behaviour of the *Activate Session Service*.

Table 144 – User Token – User Name Password Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|----------|----------------------------------|----------|
| Security | Security User Name Password | False |

6.5.124 User Token – X509 Certificate Server Facet

Table 145 describes the details of the User Token – X509 *Certificate Server* Facet. This Facet indicates that the use of an X509 certificates to identify users is supported.

Table 145 – User Token – X509 Certificate Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|----------|----------------------------------|----------|
| Security | Security User X509 | False |

6.5.125 User Token – Issued Token Server Facet

Table 146 describes the details of the User Token – Issued Token *Server* Facet. This Facet indicates that a User Token that is comprised of an issued token is supported.

Table 146 – User Token – Issued Token Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|----------|------------------------------------|----------|
| Security | Security User IssuedToken Kerberos | False |

6.5.126 User Token – Issued Token Windows Server Facet

Table 147 describes the details of the User Token – Issued Token *Windows Server* Facet. This Facet further refines the User Token – Issued Token to indicate a windows implementation of Kerberos.

Table 147 – User Token – Issued Token Windows Server Facet

| Group | Conformance Unit / Profile Title | Optional |
|----------|--|----------|
| Profile | User Token – Issued Token Facet | False |
| Security | Security User IssuedToken Kerberos Windows | False |

6.5.127 User Token – User Name Password Client Facet

Table 148 describes the details of the User Token – User Name Password *Client* Facet. This Facet defines the ability to use a user token that is comprised of a username and password.

Table 148 – User Token – User Name Password Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|----------|------------------------------------|----------|
| Security | Security User Name Password Client | False |

6.5.128 User Token – X509 Certificate Client Facet

Table 149 describes the details of the User Token – X509 *Certificate Client* Facet. This Facet defines the ability to use an X509 certificates to identify users.

Table 149 – User Token – X509 Certificate Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|----------|----------------------------------|----------|
| Security | Security User X509 Client | False |

6.5.129 User Token – Issued Token Client Facet

Table 150 describes the details of the User Token – Issued Token *Client* Facet. This Facet defines the ability to use the User Token – Issued Token (Kerberos) to connect to a *Server*.

Table 150 – User Token – Issued Token Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|----------|--|----------|
| Security | Security User IssuedToken Kerberos <i>Client</i> | False |

6.5.130 User Token – Issued Token Windows Client Facet

Table 151 describes the details of the User Token – Issued Token Windows *Client* Facet. This Facet defines the ability to use the User Token – Issued Token (Windows implementation of Kerberos) to connect to a *Server*.

Table 151 – User Token – Issued Token Windows Client Facet

| Group | Conformance Unit / Profile Title | Optional |
|----------|---|----------|
| Security | Security User IssuedToken Kerberos Windows Client | False |

6.5.131 UA-TCP UA-SC UA Binary

Table 152 describes the details of the UA-TCP UA-SC UA Binary. This transport Facet defines a combination of network protocol, security protocol and message encoding that is optimized for low resource consumption and high performance. It combines the simple TCP based network protocol UA TCP 1.0 with the binary security protocol UA SecureConversation 1.0 and the binary message encoding UA Binary 1.0.

Table 152 – UA-TCP UA-SC UA Binary

| Group | Conformance Unit / Profile Title | Optional |
|-----------------------|----------------------------------|----------|
| Protocol and Encoding | Protocol TCP Binary UA Security | False |

6.5.132 SOAP-HTTP WS-SC UA XML

Note: Deprecated in Version 1.03 because WS-SecureConversation has not been widely adopted by industry.....

6.5.133 SOAP-HTTP WS-SC UA Binary

Note: Deprecated in Version 1.03 because WS-SecureConversation has not been widely adopted by industry.....

6.5.134 SOAP-HTTP WS-SC UA XML-UA Binary

Note: Deprecated in Version 1.03 because WS-SecureConversation has not been widely adopted by industry.....

6.5.135 HTTPS UA Binary

Table 153 describes the details of the HTTPS UA Binary. This transport Facet defines a combination of network protocol, security protocol and message encoding that balances compatibility with widely used HTTPS transport and a compact UA binary encoded message for added performance. It is expected that this transport will be used to support installations where firewalls only permit HTTPS or where a WEB browser is used as *Client*. This transport requires that one of the TransportSecurity Profiles for TLS be provided.

Table 153 – HTTPS UA Binary

| Group | Conformance Unit / Profile Title | Optional |
|-----------------------|----------------------------------|----------|
| Protocol and Encoding | Protocol HTTPS with UA Binary | False |
| Security | Security TLS General | False |

6.5.136 HTTPS UA XML

Table 154 describes the details of the HTTPS UA XML. This transport Facet defines a combination of network protocol, security protocol and message encoding that uses HTTPS transport and a SOAP XML encoded message for use with standard SOAP toolkits. This transport requires that one of the TransportSecurity Profiles for TLS be provided.

Table 154 – HTTPS UA XML

| Group | Conformance Unit / Profile Title | Optional |
|-----------------------|----------------------------------|----------|
| Protocol and Encoding | Protocol HTTPS with Soap | False |
| Security | Security TLS General | False |

6.5.137 Security User Access Control Full

Table 155 describes the details of the Security User Access Control Full. A *Server* that supports this profile supports restricting multiple levels of access to all *Nodes* in the *AddressSpace* based on the validated user.

Table 155 – Security User Access Control Full

| Group | Conformance Unit / Profile Title | Optional |
|---------------------|--------------------------------------|----------|
| Profile | Security User Access Control Base | False |
| Address Space Model | Address Space User Access Level Full | False |

6.5.138 Security User Access Control Base

Table 156 describes the details of the Security User Access Control Base. A *Server* that supports this profile supports restricting some level of access to some *Nodes* in the *AddressSpace* based on the validated user.

Table 156 – Security User Access Control Base

| Group | Conformance Unit / Profile Title | Optional |
|---------------------|--|----------|
| Address Space Model | Address Space User Access Level Base | False |
| Security | Security User IssuedToken Kerberos | True |
| Security | Security User IssuedToken Kerberos Windows | True |
| Security | Security User Name Password | False |
| Security | Security User X509 | True |

6.5.139 Security Time Synchronization

Table 157 describes the details of the Security Time Synchronization. This Facet indicates that the application supports the minimum required level of time synchronization to ensure secure communication. One of the optional time synchronization conformance units must be supported.

Table 157 – Security Time Synchronization

| Group | Conformance Unit / Profile Title | Optional |
|----------|--|----------|
| Security | Security Time Synch – Configuration | False |
| Security | Security Time Synch – NTP / OS Based support | True |
| Security | Security Time Synch – UA based support | True |

6.5.140 Best Practice – Audit Events

Table 158 describes the details of the Best Practice – Audit Events. Subscriptions for Audit Events shall be restricted to authorized personnel.

Table 158 – Best Practice – Audit Events

| Group | Conformance Unit / Profile Title | Optional |
|---------------|----------------------------------|----------|
| Miscellaneous | Best Practice – Audit Events | False |

6.5.141 Best Practice – Alarm Handling

Table 159 describes the details of the Best Practice – *Alarm* Handling. A *Server* should restrict critical alarm handling functionality to users that have the appropriate rights to perform these actions

Table 159 – Best Practice – Alarm Handling

| Group | Conformance Unit / Profile Title | Optional |
|---------------|----------------------------------|----------|
| Miscellaneous | Best Practice – Alarm Handling | False |

6.5.142 Best Practice – Random Numbers

Table 160 describes the details of the Best Practice – Random Numbers. All random numbers that are required for security should use appropriate cryptographic library based random number generators.

Table 160 – Best Practice – Random Numbers

| Group | Conformance Unit / Profile Title | Optional |
|---------------|----------------------------------|----------|
| Miscellaneous | Best Practice – Random Numbers | False |

6.5.143 Best Practice – Timeouts

Table 161 describes the details of the Best Practice – Timeouts. The administrator should be able to configure reasonable timeouts for Secure Channels, *Sessions* and *Subscriptions*. Setting these timeouts allows limiting Denial of Service attacks and overload issues.

Table 161 – Best Practice – Timeouts

| Group | Conformance Unit / Profile Title | Optional |
|---------------|----------------------------------|----------|
| Miscellaneous | Best Practice – Timeouts | False |

6.5.144 Best Practice – Administrative Access

Table 162 describes the details of the Best Practice – Administrative Access. The *Server* and *Client* allow restricting the use of certain *Services* and access to parts of the *AddressSpace* to administrative personnel. This includes multiple level of administrative access on platforms that support multiple administrative roles (such as Windows or Linux).

Table 162 – Best Practice – Administrative Access

| Group | Conformance Unit / Profile Title | Optional |
|---------------|---------------------------------------|----------|
| Miscellaneous | Best Practice – Administrative Access | False |

6.5.145 Best Practice – Strict Message Handling

Table 163 describes the details of the Best Practice – Strict Message Handling. *Server* and *Client* reject messages that are incorrectly formed as specified in Part 4 and Part 6.

Table 163 – Best Practice – Strict Message Handling

| Group | Conformance Unit / Profile Title | Optional |
|---------------|---|----------|
| Miscellaneous | Best Practice – Strict Message Handling | False |

6.5.146 Best Practice – Audit Events Client

Table 164 describes the details of the Best Practice – Audit Events *Client*. Audit Tracking system connect to a *Server* using a secure channel and under the appropriate authorization to allow access to Audit events.

Table 164 – Best Practice – Audit Events Client

| Group | Conformance Unit / Profile Title | Optional |
|---------------|-------------------------------------|----------|
| Miscellaneous | Best Practice – Audit Events Client | False |

6.5.147 SecurityPolicy – None

Table 165 describes the details of the SecurityPolicy – None. This security Facet defines a SecurityPolicy used for configurations with the lowest security needs. This SecurityPolicy can affect the behaviour of the CreateSession and Activate *Session* services. It also results in a SecureChannel which has no Channel Security. By default this SecurityPolicy should be disabled if any other SecurityPolicies are available.

Table 165 – SecurityPolicy – None

| Group | Conformance Unit / Profile Title | Optional |
|----------|---|----------|
| Security | Security None | False |
| Security | Security None CreateSession ActivateSession | False |
| Security | Security None CreateSession ActivateSession 1.0 | True |

6.5.148 SecurityPolicy – Basic128Rsa15

Table 166 describes the details of the SecurityPolicy – Basic128Rsa15. This security Facet defines a Security Policy for configurations with medium security. It requires a PKI infrastructure.

As computing power increases, SecurityPolicies are expected to expire. NIST provides guidelines for expected expiration dates for individual algorithms. These guidelines provided recommended dates at which the algorithm should be replaced or upgraded to a more secure algorithm. They do not indicate a failure of the algorithm. NIST recommends users of this SecurityPolicy should consider upgrading it for key lengths less than 2048 in 2010. NIST also recommends that this SecurityPolicy should be deprecated in 2012 for key lengths less than 2048. It is recommended that *Servers* and *Client* support all security profiles and developers provide the recommended profile as a default. It is up to an administrator to configure the actual exposed SecurityPolicies.

Table 166 – SecurityPolicy – Basic128Rsa15

| Group | Conformance Unit / Profile Title | Optional |
|----------|----------------------------------|----------|
| Security | Security Basic 128Rsa15 | False |
| Security | Security Certificate Validation | False |
| Security | Security Encryption Required | False |
| Security | Security Signing Required | False |

6.5.149 SecurityPolicy – Basic256

Table 167 describes the details of the SecurityPolicy – Basic256. This security Facet defines a Security Policy for configurations with medium to high security needs. It requires a PKI infrastructure.

As computing power increases, SecurityPolicies are expected to expire. NIST provides guidelines for expected expiration dates for individual algorithms. These guidelines provided recommended dates at which the algorithm should be replaced or upgraded to a more secure algorithm. They do not indicate a failure of the algorithm. NIST recommends users of this SecurityPolicy should consider upgrading it for key sizes less than 2048 in 2010. NIST also recommends that this SecurityPolicy should be deprecated in 2012 for key sizes less than 2048. It is recommended that *Servers* and *Client* support all security profiles and developers provide the recommended profile as a default. It is up to an administrator to configure the actual exposed SecurityPolicies.

Table 167 – SecurityPolicy – Basic256

| Group | Conformance Unit / Profile Title | Optional |
|----------|----------------------------------|----------|
| Security | Security Basic 256 | False |
| Security | Security Certificate Validation | False |
| Security | Security Encryption Required | False |
| Security | Security Signing Required | False |

6.5.150 SecurityPolicy – Basic256Sha256

Table 168 describes the details of the SecurityPolicy – Basic256Sha256. This security Facet defines a Security Policy for configurations with high security needs. It requires a PKI infrastructure.

As computing power increases, SecurityPolicies are expected to expire. NIST provides guidelines for expected expiration dates for individual algorithms. These guidelines provided recommended dates at which the algorithm should be replaced or upgraded to a more secure algorithm. They do not indicate a failure of the algorithm. This security Policy has no published end dates as of this time. It is recommended that *Servers* and *Client* support all security profiles and developers provide the recommended profile as a default. It is up to an administrator to configure the actual exposed SecurityPolicies.

Table 168 – SecurityPolicy – Basic256Sha256

| Group | Conformance Unit / Profile Title | Optional |
|----------|----------------------------------|----------|
| Security | Security Basic 256 Sha256 | False |

6.5.151 TransportSecurity – TLS 1.0

Note: Deprecated in Version 1.03 because the RC4 algorithm is not considered secure anymore.

6.5.152 TransportSecurity – TLS 1.1

Note: Deprecated in Version 1.03 because the RC4 algorithm is not considered secure anymore.

6.5.153 TransportSecurity – TLS 1.2

Table 169 describes the details of the SecurityPolicy – TLS 1.2. This Facet defines a transport security for configurations with high security needs. It makes use of TLS 1.2 and uses TLS_RSA_WITH_AES_256_CBC_SHA256.

As computing power increases, security algorithms are expected to expire. NIST provides guidelines for expected expiration dates for individual algorithms. These guidelines provide recommended dates at which the algorithm should be replaced or upgraded to a more secure algorithm. They do not indicate a failure of the algorithm. NIST has no recommendations for this TransportSecurity. It is recommended that *Servers* and *Client* support all security profiles

and developers provide the recommended profile as a default. It is up to an administrator to configure the actual exposed TransportSecurity Profiles.

Table 169 – TransportSecurity – TLS 1.2

| Group | Conformance Unit / Profile Title | Optional |
|----------|---|----------|
| Security | Security TLS_RSA_WITH_AES_256_CBC_SHA256 | False |

6.5.154 TransportSecurity – TLS 1.2 with PFS

Table 170 describes the details of the SecurityPolicy – TLS 1.2 with PFS. This Facet defines a transport security for configurations with high security needs and perfect forward security (PFS). It makes use of TLS 1.2 and uses TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 or TLS_DHE_RSA_WITH_AES_256_CBC_SHA256. As computing power increases, security algorithms are expected to expire. NIST provides guidelines for expected expiration dates for individual algorithms. These guidelines provide recommended dates at which the algorithm should be replaced or upgraded to a more secure algorithm. They do not indicate a failure of the algorithm. NIST has no recommendations for this TransportSecurity. It is recommended that Servers and Clients support all security profiles and developers provide the recommended profile as a default. It is up to an administrator to configure the actual exposed TransportSecurity Profiles.

Table 170 – TransportSecurity – TLS 1.2 with PFS

| Group | Conformance Unit / Profile Title | Optional |
|----------|---|----------|
| Security | Security TLS_DHE_RSA_WITH_AES_nnn_CBC_SHA256 | False |

Bibliography

Test Specifications

Compliance Part 8 UA Server: *OPC Test Lab Specification – Part 8 – UA Server*

Compliance Part 9 UA Client: *OPC Test Lab Specification – Part 9 – UA Client*
