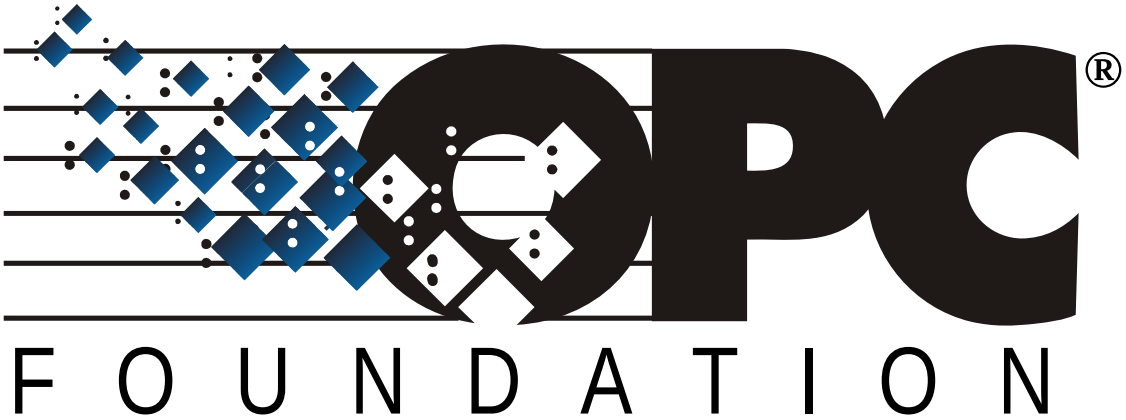


# OPC Unified Architecture

# Specification

# Part 7: Profiles

# Release 1.00

# February 6, 2009

| | | | |
|---|---|---|---|
| Specification Type: | Industry Standard Specification | Comments: | Report or view errata: http://www.opcfoundation.org/errata |
| Title: | OPC Unified Architecture<br><br>Part 7 :Profiles | Date: | February 6, 2009 |
| Version: | Release 1.00 | Software: | MS-Word |
| | | Source: | OPC UA Part 7 - Profiles 1.00 Specification.doc |
| Author: | OPC Foundation | Status: | Release |

# CONTENTS

Page

## FIGURES

**TABLES**

# OPC FOUNDATION

# UNIFIED ARCHITECTURE –

## FOREWORD

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



## AGREEMENT OF USE

COMPLIANCE

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

TRADEMARKS

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

GENERAL PROVISIONS

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

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

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

ISSUE REPORTING

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

# OPC Unified Architecture Specification

# Part 7: *Profiles*

## 1 Scope

This specification describes the OPC Unified Architecture (OPC UA) *Profiles*. The *Profiles* in this part are used to describe the functionality that an OPC UA *Server* exposes or that an OPC UA *Client* consumes. The details of the functionality are specified in other part of the OPC UA specification.

*Profiles* are used by vendors to advertise the OPC UA capabilities of their products. The *Profiles* a product supports will typically appear on product data sheets. Buyers will use this *Profile* information to specify and purchase products that work together and meet specific application requirements. Most OPC UA applications will conform to several, but not all of the *Profiles*.

A product's conformance to a *Profile* is measured using self-testing tools (Compliance Test Tools) and via Independent Certification Test Labs. Some *Profiles* describe functionality that is only testable in a test lab environment. These *Profiles* will clearly state this in their description.

## 2 Reference documents

Part 1 : OPC UA Specification: Part 1 – Concepts, Version 1.01 or later.
http://www.opcfoundation.org/UA/Part1/

Part 2 : OPC UA Specification: Part 2 – Security Model, Version 1.01 or later
http://www.opcfoundation.org/UA/Part2/

Part 3 : OPC UA Specification: Part 3 – Address Space Model, Version 1.01 or later
http://www.opcfoundation.org/UA/Part3/

Part 4 : OPC UA Specification: Part 4 – Services, Version 1.01 or later
http://www.opcfoundation.org/UA/Part4/

Part 5 : OPC UA Specification: Part 5 – Information Model, Version 1.01 or later
http://www.opcfoundation.org/UA/Part5/

Part 6 : OPC UA Specification: Part 6 – Mappings, Version 1.0 or later
http://www.opcfoundation.org/UA/Part6/

Part 8 : OPC UA Specification: Part 8 – Data Access, Version 1.01 or later
http://www.opcfoundation.org/UA/Part8/

Part 9 : OPC UA Specification: Part 9 – Alarms and Conditions, Version 1.0 or later
http://www.opcfoundation.org/UA/Part9/

Part 10 : OPC UA Specification: Part 10 – Programs, Version 1.01 or later
http://www.opcfoundation.org/UA/Part10/

Part 11 : OPC UA Specification: Part 11 – Historical Access, Version 1.01 or later
http://www.opcfoundation.org/UA/Part11/

Part 12 : OPC UA Specification: Part 12 – Discovery, Version 1.0 or later
http://www.opcfoundation.org/UA/Part12/

Part 13 : OPC UA Specification: Part 13 – Aggregates, Version 1.0 or later
http://www.opcfoundation.org/UA/Part13/

Test Specifications

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

http://www.opcfoundation.org/

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

http://www.opcfoundation.org/

Compliance Part 10] : OPC Test Lab Specification: Part 10 – UA Abstract Test

http://www.opcfoundation.org/

Kerberos Token : wss-v1.1-spec-os-KerberosTokenProfile

http://www.oasis-open.org/committees/download.php/16788/wss-v1.1-spec-os-KerberosTokenProfile.pdf

# 3 Terms, definitions, and conventions

## 3.1 OPC UA part 1 terms

The following terms defined in Part 1 apply.

1) AddressSpace
2) Alarm
3) Attribute
4) Certificate
5) Client
6) Complex Data
7) Condition
8) Discovery
9) Event
10) EventNotifier
11) Message
12) Method
13) MonitoredItem
14) Node
15) NodeClass
16) Notification
17) Object

18) Object Type
19) Profile
20) Program
21) Reference
22) ReferenceType
23) Server
24) Service
25) Service Set
26) Session
27) Subscription
28) Variable
29) View

### 3.2 OPC UA part 2 terms

The following terms defined in Part 2 apply.

Asymmetric Encryption
Asymmetric Signature
Auditing
Authentication
Authorization
Message Signature
Private Key
Public Key
Public Key Infrastructure (PKI)
Rivest-Shamir-Adleman (RSA)
Secure Channel
Symmetric Cryptography
Symmetric Encryption
Symmetric Signature
30) X.509 Certificate

### 3.3 OPC UA part 3 terms

The following terms defined in Part 3 apply.

1) DataVariable
2) EventType
3) HierarchicalReference
4) ModellingRule
5) Property
6) VariableType

### 3.4 OPC UA part 4 terms

The following terms defined in Part 4 apply.

1) Deadband
2) EndPoint
3) SoftwareCertificate

### 3.5 OPC UA part 6 terms

The following terms defined in Part 6 apply.

1) Data Encoding
2) Stack Profile
3) Transport Protocol

### 3.6 OPC UA part 8 terms

The following terms defined in Part 8 apply.

1) AnalogItem
2) DataItem
3) DiscreteItem
4) EngineeringUnits

### 3.7 OPC UA Profile terms

#### 3.7.1 ConformanceUnit

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

### 3.8 Abbreviations and symbols

| | |
|---|---|
| API | Application Programming Interface |
| DA | Data Access |
| HA | Historical Access |
| HMI | Human Machine Interface |
| UA | Unified Architecture |

## 4 Overview

### 4.1 General

The OPC UA multipart specification 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 of the features they support. This part 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 test cases. The figure also illustrates a feature of the OPC UA

compliance test tool, 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.

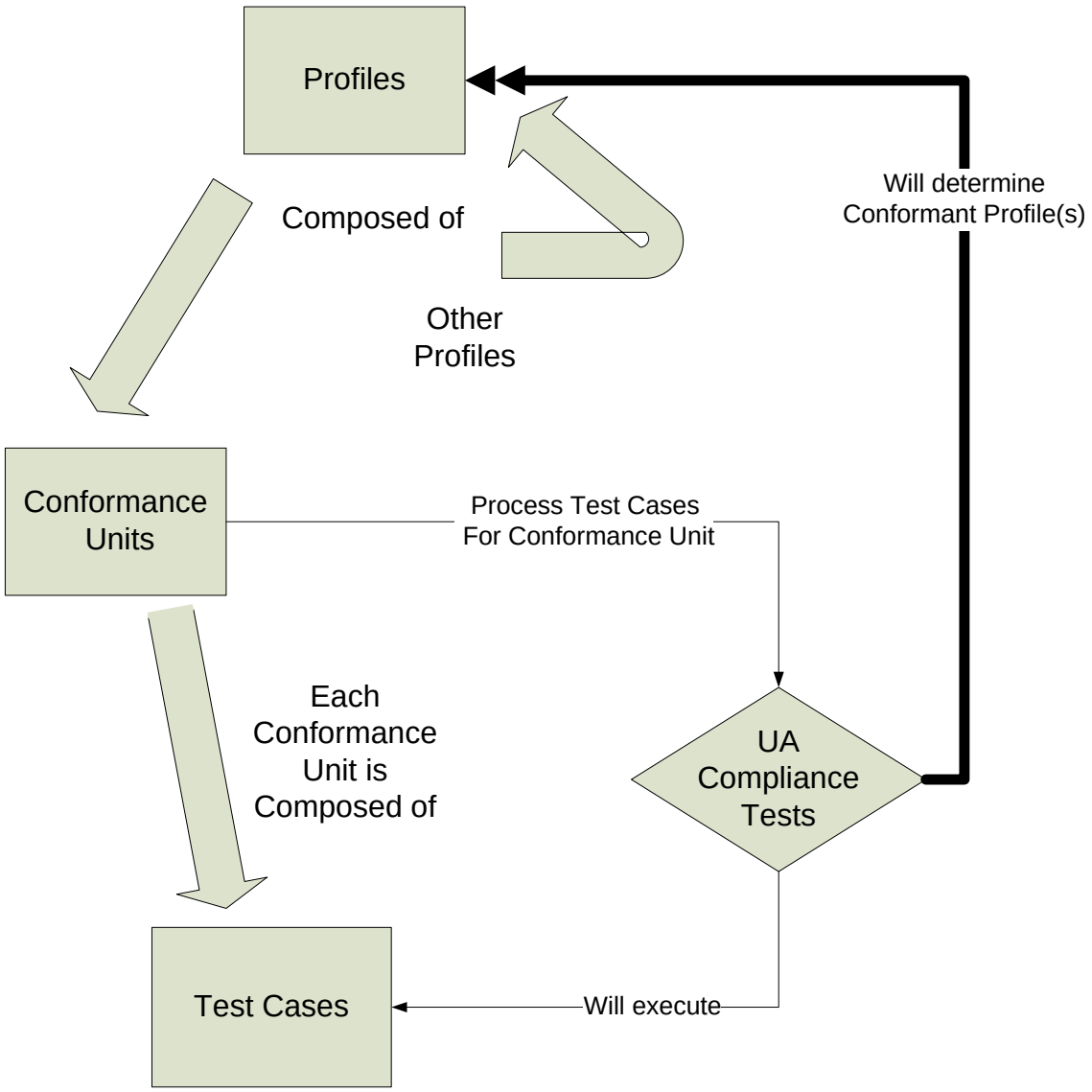


**Figure 1 - Profile - ConformanceUnit - Test cases**

### 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. For each ConformanceUnit there are a number of test cases that test the functionality described in 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 either the appropriate part of the OPC UA specification or the details provided in the OPC UA Client or OPC UA Server test specification.

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

### 4.3 Profiles

Profiles are named groupings of ConformanceUnits. The Servers and Clients in an OPC UA application will provide the names of Profiles that they support. The definition of Profiles is a dynamic activity, in that it is expected that new Profiles will be added in the future. A Profile can be defined to inherit from an existing Profile. The new Profile may add additional ConformanceUnits. These additional ConformanceUnits may add additional features that are to be tested. The additional ConformanceUnits may also further restrict inherited ConformanceUnits.

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 following some conventions see clause 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. Categories include: Server and Client, but they could also include Power Generation or Chemical Plant. A Profile can be assigned to more than one category.

Table 1 – Profiles Categories contains the list of defined Profile categories.

**Table 1 – Profiles Categories**

| Category | Description |
|---|---|
| Client | Profiles of this category specify a complete functional set for an OPC UA Client. The URI of such profiles has to be part of a Software Certificate passed in the CreateSession request. |
| Security | Profiles of this category specify a security policy. The URI of such profiles has to be part of an Endpoint Description returned from the GetEndpoint service. |
| Server | Profiles of this category specify a complete functional set for an OPC UA Server. The URI of such profiles has to be part of a Software Certificate returned with the CreateSession service response. |
| Transport | Profiles of this category specify a specific protocol mapping. The URI of such profiles has to be part of an Endpoint Description. |

## 5 ConformanceUnits

### 5.1 Overview

A ConformanceUnit represents an individually testable entity. The list of ConformanceUnits is large and as a result ConformanceUnits are combined for easier understanding into named conformance groups. These groups closely correspond to the Service Sets in Part 4 and to the OPC UA information models. Table 2 provides a list of the conformance groups. These groups and the ConformanceUnits that they describe are detailed in the following sections. Conformance Groups have no impact on compliance testing; they are used only for organizational reasons in this document.

**Table 2 - Conformance Groups**

| Group | Description |
|---|---|

| Group | Description |
|---|---|
| Address Space Model | Defines ConformanceUnits for various features of the OPC UA AddressSpace. |
| Alarms & Conditions | All ConformanceUnits that are associated with the enhanced OPC UA information model for Conditions, acknowledgeable Conditions, confirmations and Alarms. |
| Attribute Services | Includes ConformanceUnits to Read or Write current or historical Attribute values. |
| Auditing | User level security includes support for security audit trails, with traceability between Client and Server audit logs. |
| Base Information | All information elements as defined in Part 5 of the OPC UA specification. |
| Data Access | ConformanceUnits specific to Clients and Servers that deal with the representation and use of automation data as specified in Part 8 of the OPC UA specification. |
| Discovery Services | ConformanceUnits with focus on Server Endpoint Discovery |
| Historical Access | Access to archived data of node Attribute values or Events. |
| Method Services | Methods represent the function calls of Objects. Methods are invoked and return only after completion (successful or unsuccessful). |
| Monitored Item Services | Clients define MonitoredItems to subscribe to data and Events. Each MonitoredItem identifies the item to be monitored and the Subscription to use to send Notifications. |
| Node Management Services | Bundles ConformanceUnits for all Services to add and delete AddressSpace Nodes and References. |
| Programs | Programs are complex functions in a server or underlying system that can be invoked and managed by an OPC UA Client. Programs can represent any level of functionality within a system or process in which client control or intervention is required and progress monitoring is desired |
| Protocol and Encoding | Covers all transport and encoding combinations that are specified in Part 6 of the OPC UA specification. |
| Query *Services* | A Query may be used to return a subset of data from the View. |
| Redundancy | The design of OPC UA ensures that vendors can create redundant *Clients* and redundant *Servers* in a consistent manner. Redundancy may be used for high availability, fault tolerance and load balancing. |
| Security | Security related *ConformanceUnits* that can be profiled. |
| *Session Services* | An (OPC UA) *Session* is an application layer connection. |
| *Subscription Services* | Subscriptions are used to report *Notifications* to the *Client*. |
| View *Services* | *Clients* use the View *Service* Set to navigate through the *AddressSpace* or through a View as subset of the *AddressSpace*. |

### 5.2 Services

The following tables 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 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 test cases. These test cases can only further restrict the existing test cases. An example would be one in which the number of connections is tested, where the test case of the parent required at least one connection and the derived *ConformanceUnit* would require a test case 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 | Derived |
|---|---|---|---|
| *Server* | *Discovery* Get Endpoints | Support the GetEndpoints *Service* to obtain all Endpoints of one *Server*.<br>This includes filtering based on *Profiles*. | |
| *Server* | *Discovery* Find *Servers* Self | Support the FindServers *Service* only for itself. | |
| *Server* | *Discovery* Register | Call the RegisterServers *Service* to register itself (OPC UA *Server*) with an external *Discovery Service* via a secure channel with a securityMode other than "None". | |
| *Server* | *Discovery* Configuration | Allow configuration of the *Discovery Server* URL where the *Server* will register itself.<br>Allow complete disabling of registration with a *Discovery Server*. | |
| *Client* | *Discovery Client* Find *Servers* Basic | Uses the FindServers *Service* to obtain all *Servers* installed on a given node. | |
| *Client* | *Discovery Client* Find *Servers* with URI | Use FindServers *Service* to obtain URLs for specific *Server* URIs. | |
| *Client* | *Discovery Client* Find *Servers* Dynamic | Detect new *Servers* after an initial FindServers *Service* call. | |
| *Client* | *Discovery Client* Get Endpoints Basic | Uses the GetEndpoints *Service* to obtain all Endpoints for a given *Server* URI. | |
| *Client* | *Discovery Client* Get Endpoints Dynamic | Detect changes to the Endpoints after an initial GetEndpoints *Service* call. | |
| *Client* | *Discovery Client* Configure Endpoint | Allow specification of an Endpoint without going through the *Discovery Service* Set. | |

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 | Derived |
|---|---|---|---|
| *Server* | *Session* General *Service* Behaviour | Implement basic *Service* behaviour. This includes in particular:<br>- checking the authentication token<br>- returning the requestHandle in responses<br>- returning available diagnostic information as requested with the 'returnDiagnostics' parameter<br>- respecting a timeoutHint | |

| Category | Title | Description | Derived |
|---|---|---|---|
| *Server* | *Session* Base | Support the *Session Service* Set (CreateSession, ActivateSession, CloseSession) except the use of ActivateSession to change the *Session* user. This includes correct handling of all parameters that are provided.<br><br>Note that for the CreateSession and ActivationSession services, if the SecurityMode = None then:<br>1) The Application *Certificate* and Nonce are optional.<br>2) The signatures are null/empty. | |
| *Server* | *Session* Cancel | Support the Cancel *Service* to cancel outstanding requests. | |
| *Server* | *Session* Minimum 10 Parallel | Support minimum 10 parallel Sessions (total for all *Clients*). | |
| *Server* | *Session* Minimum 50 Parallel | Support minimum 50 parallel Sessions (total for all *Clients*). | *Session* Minimum 10 Parallel |
| *Client* | *Session Client* General *Service* Behaviour | Implement basic *Service* behaviour. This includes in particular:<br>- including the proper authentication token of the *Session*<br>- creating requestHandles if needed<br>- requesting diagnostic information with the 'returnDiagnostics' parameter<br>- evaluate the serviceResult and operational results | |
| *Client* | *Session Client* Base | Use the *Session Service* Set (CreateSession, ActivateSession, CloseSession) except the use of ActivateSession to change the *Session* user. This includes correct handling of all parameters that are provided<br>Note that for the CreateSession and ActivationSession services, if the SecurityMode = None then:<br>1) The Application *Certificate* and Nonce are optional.<br>2) The signatures are null/empty. | |
| *Client* | *Session Client* Renew NodeIds | This *ConformanceUnit* applies to *Clients* that persist NodeIds.<br>Verify that the Namespace Table has not changed for NodeIds that the *Client* has persisted and is going to re-use beyond a *Session* lifetime. If changes occurred the *Client* has to recalculate the Namespace Indices of the respective NodeIds. | |
| *Client* | *Session Client* Impersonate | Use of ActivateSession to change the *Session* user (impersonation). | |
| *Client* | *Session Client* KeepAlive | Make periodic requests to keep the *Session* alive. | |
| *Client* | *Session Client* Detect Shutdown | Read or monitor the ServerStatus/State *Variable* to recognize a potential shutdown of the *Server* and clean up resources. | |
| *Client* | *Session Client* Cancel | Use the Cancel *Service* to cancel outstanding requests. | |

| Category | Title | Description | Derived |
|---|---|---|---|
| *Client* | *Session Client* Auto Reconnect | Automatic *Client* reconnect including:<br>- ActivateSession with new SecureChannel if SecureChannel is not longer valid but *Session* is still valid<br>- Creation of a new *Session* only if *Session* is not longer valid | |

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 | Derived |
|---|---|---|---|
| *Server* | *Node* Management Add *Node* | Support the AddNodes service to add one or more *Nodes* into the *AddressSpace*. | |
| *Server* | *Node* Management Delete *Node* | Support the DeleteNodes *Service* to delete one or more *Nodes* from the *AddressSpace*. | |
| *Server* | *Node* Management Add Ref | Support the AddReferences *Service* to add one or more *References* to one or more *Nodes*. | |
| *Server* | *Node* Management Delete Ref | Support the DeleteReferences *Service* to delete one or more *References* of a *Node*. | |
| *Client* | *Node* Management *Client* | Use *Node* Management *Services* to add or delete *Nodes* and to add or delete *References* in an OPC UA *Server*'s *AddressSpace*. | |

The View *Service* Set is composed of a multiple *ConformanceUnits* (see Table 6). All *Servers* support some aspects this conformance group.

**Table 6 – View Services**

| Category | Title | Description | Derived |
|---|---|---|---|
| *Server* | View Basic | Support the View *Service* Set (Browse, BrowseNext). | |
| *Server* | View TranslateBrowsePath | Support TranslateBrowsePathsToNodeIds *Service*. | |
| *Server* | View RegisterNodes | Support the RegisterNodes and UnregisterNodes *Services* as a way to optimize access to repeatedly used *Nodes* in the *Server*'s *AddressSpace*. | |
| *Server* | View Minimum Continuation Point 01 | Support minimum 1 continuation point per *Session*. | |
| *Server* | View Minimum Continuation Point 05 | Support minimum 5 continuation points per *Session*.<br>This number has to be supported for at least half of the minimum required sessions. | View Minimum Continuation Point 01 |
| *Client* | View *Client* Basic Browse | Use Browse and BrowseNext *Services* to navigate through the *Server*'s *AddressSpace*. Make use of the referenceTypeId and the nodeClassMask to specify the needed *References*. | |

| Category | Title | Description | Derived |
|---|---|---|---|
| *Client* | View *Client* Basic ResultSet Filtering | Make use of the resultMask parameter to optimize the result set to be returned by the *Server*. | |
| *Client* | View *Client* TranslateBrowsePath | Use the TranslateBrowsePathsToNodeIds *Service* to identify the NodeIds for *Nodes* where a starting *Node* and a BrowsePath is known. Make bulk operations rather than multiple calls whenever possible. | |
| *Client* | View *Client* RegisterNodes | Use the RegisterNodes *Service* to optimize access for *Nodes* that are used repeatedly. Use UnregisterNodes when *Nodes* are not used anymore. | |

The Query *Service* Set is composed of multiple *ConformanceUnits* (see Table 7). *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 7 – Query Services**

| Category | Title | Description | Derived |
|---|---|---|---|
| | Undefined | No Conformance Units defined | |

The *Attribute Service* Set is composed of multiple *ConformanceUnits* (see Table 8). The majority of the *Attribute* service set is a core functionality of the OPC UA specification.

**Table 8 – Attribute Services**

| Category | Title | Description | Derived |
|---|---|---|---|
| *Server* | *Attribute* Read | Support the Read *Service* to read one or more *Attributes* of one or more *Nodes*. This includes support of the IndexRange parameter to read a single element or a range of elements when the *Attribute* value is an array | |
| *Server* | *Attribute* Read Complex | Support reading and encoding ComplexData (structures). | |
| *Server* | *Attribute* Write Values | Support the Write *Service* to write values to one or more *Attributes* of one or more *Nodes*. | |
| *Server* | *Attribute* Write Complex | Support writing and decoding ComplexData. | |
| *Server* | *Attribute* Write StatusCode & TimeStamp | Support writing of StatusCode and Timestamps. | |
| *Server* | *Attribute* Write Index | Support the IndexRange to write a single element or a range of elements when the *Attribute* value is an array. | |
| *Server* | *Attribute* Alternate Encoding | Support alternate Data Encoding when reading value *Attributes*.<br>By default, every *Server* has to support the Data Encoding of the currently used Stack *Profile* (i.e. binary with UA Binary Encoding and XML with XML Encoding). This *ConformanceUnit* - when supported - specifies that the other Data Encoding is supported in addition. | |

| Category | Title | Description | Derived |
|---|---|---|---|
| *Client* | *Attribute Client* Read Base | Use the Read *Service* to read one or more *Attributes* of one or more *Nodes*. This includes use of an IndexRange to select a single element or a range of elements when the *Attribute* value is an array.<br>*Clients* shall use bulk operations whenever possible to reduce the number of *Service* invocations. | |
| *Client* | *Attribute Client* Read with proper Encoding | This *ConformanceUnit* refers to the ability of a *Server* to support more than one Data Encoding for *Attribute* values. *Clients* can discover the available encodings and can explicitely choose one when calling the Read *Service*. | |
| *Client* | *Attribute Client* Read Complex | Read and decode ComplexData (structures). | |
| *Client* | *Attribute Client* Write Base | Use the Write *Service* to write values to one or more *Attributes* of one or more *Nodes*. This includes use of an IndexRange to select a single element or a range of elements when the *Attribute* value is an array.<br>*Clients* shall use bulk operations whenever possible to reduce the number of *Service* invocations. | |
| *Client* | *Attribute Client* Write Complex | Write and Encode ComplexValues (structures). | |
| *Client* | *Attribute Client* Write Quality & TimeStamp | Use the Write *Service* to also write StatusCode and/or Timestamps. | |

The *Method Service* Set is composed of *ConformanceUnits* (see Table 9). 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 9 – Method Services**

| Category | Title | Description | Derived |
|---|---|---|---|
| *Server* | *Method* Call | Support the Call *Service* to call (invoke) a *Method*. | |
| *Client* | *Method Client* Call | Use the Call *Service* to call one or several Methods. | |

The *MonitoredItem 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 – Monitored Item Services**

| Category | Title | Description | Derived |
|---|---|---|---|
| *Server* | Monitor Basic | Support the following *MonitoredItem Services*: CreateMonitoredItems, ModifyMonitoredItems, DeleteMonitoredItems, SetMonitoringMode. | |
| *Server* | Monitor Value Change | Support creation of *MonitoredItems* for *Attribute* value changes. This includes support of the IndexRange to select a single element or a range of elements when the *Attribute* value is an array. | |

| Category | Title | Description | Derived |
|---|---|---|---|
| *Server* | Monitor Alternate Encoding | Support alternate encoding when monitoring value *Attributes*.<br>By default, every *Server* has to support the encoding of the currently used Stack *Profile* ( i.e. binary with UA Binary Encoding and XML with XML Encoding). This *ConformanceUnit* - when supported - specifies that the other encoding is supported in addition. | |
| *Server* | Monitor Items 10 | Support at least 10 *MonitoredItems* per *Subscription*. | |
| *Server* | Monitor Items 100 | Support at least 100 *MonitoredItems* per *Subscription*.<br>This number has to be supported for at least half of the required Subscriptions for half of the required Sessions. | Monitor Items 10 |
| *Server* | Monitor Items 500 | Support at least 500 *MonitoredItems* per *Subscription*.<br>This number has to be supported for at least half of the required Subscriptions for half of the required Sessions. | Monitor Items 100 |
| *Server* | Monitor QueueSize_1 | This *ConformanceUnit* does not require queueing when multiple value changes occur during a publish period.<br>I.e. the latest change will be sent in the *Notification*. | |
| *Server* | Monitor MinQueueSize_02 | Support at least 2 queue entries for *MonitoredItems*.<br>*Servers* often will adapt the queue size to the number of currently *MonitoredItems*. However, it is expected that *Servers* support this minimum queue size for at least one third of the supported *MonitoredItems*. | |
| *Server* | Monitor Triggering | Support the SetTriggering *Service* to create and/or delete triggering links for a triggering item. | |
| *Server* | Monitor Events | Support creation of *MonitoredItems* for a "Notifier *Attribute*" for the purpose of *Event Notification*. | |
| *Server* | Monitor Complex *Event* Filter | Support for complex *Event* filters, where complex is defined as supporting the complex filter operands | |
| *Client* | Monitor *Client* Value Change | Use the *MonitoredItem Service* Set to register items for changes in *Attribute* value.<br>Use CreateMonitoredItems to register the *Node*/*Attribute* tuple. Set proper sampling interval, Deadband filter and queueing mode.<br>Use disabling / enabling instead of deleting and re-creating a *MonitoredItem*.<br>Use bulk operations rather than individual service requests to reduce communication overhead. | |
| *Client* | Monitor *Client* by Index | Use the IndexRange to select a single element or a range of elements when the *Attribute* value is an array. | |
| *Client* | Monitor *Client* Events | Use the *MonitoredItem Service* Set to create *MonitoredItems* for *Event* notifications. | |
| *Client* | Monitor *Client Event* Filter | Use the *Event* filter when calling CreateMonitoredItems to filter the desired Events and to select the columns to be provided for each *Event Notification*. | |

| Category | Title | Description | Derived |
|---|---|---|---|
| *Client* | Monitor *Client* Complex Evt Filter | Uses complex *Event* filters | |
| *Client* | Monitor *Client* Modify | Use ModifyMonitoredItems *Service* to change the configuration setting.<br>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.<br>Use proper monitoring mode for these items.<br>Use SetTriggering *Service* to link these items to the trigger item. | |

The *Subscription Service* Set is composed of multiple *ConformanceUnits* (see Table 11). *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 11 – Subscription Services**

| Category | Title | Description | Derived |
|---|---|---|---|
| *Server* | *Subscription* Basic | Support the following *Subscription Services*: CreateSubscription, ModifySubscription, DeleteSubscriptions, Publish, Republish, SetPublishingMode. | |
| *Server* | *Subscription* Minimum 02 | Support at least 2 Subscriptions per *Session*. | |
| *Server* | *Subscription* Minimum 05 | Support at least 5 Subscriptions per *Session*.<br>This number has to be supported for at least half of the minimum required sessions. | *Subscription* Minimum 02 |
| *Server* | *Subscription* Publish Min 05 | Support at least 5 Publish *Service* requests per *Session*. | |
| *Server* | *Subscription* Publish Min 10 | Support at least 10 Publish *Service* requests per *Session*.<br>This number has to be supported for at least half of the minimum required sessions. | *Subscription* Publish Min 05 |
| *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. | |
| *Client* | *Subscription Client* Basic | Use the *Subscription* and *MonitoredItem Service* Set as an efficient means to detect changes of *Attribute* values and / or to receive *Event* occurrences.<br>Set appropriate intervals for publishing, keep alive notifications and total *Subscription* lifetime.<br>Supply a sufficient number of Publish requests to the *Server* so that *Notifications* can be sent whenever a publish timer expires.<br>Acknowledge received *Notifications* with subsequent Publish requests. | |
| *Client* | *Subscription Client* Republish | Evaluate the sequence number in *Notifications* to detect lost *Notifications*.<br>Use Republish to request missing *Notifications*. | |

| Category | Title | Description | Derived |
|---|---|---|---|
| *Client* | *Subscription Client* Modify | Allow modification of the *Subscription* configuration using the ModifySubscription *Service*. | |
| *Client* | *Subscription Client* Multiple | Use multiple Subscriptions to reduce the payload of individual *Notifications*. | |
| *Client* | *Subscription Client* Publish Configurable | Send multiple Publish *Service* requests to assure that the *Server* is always able to send *Notifications*.<br>The number of parallel Publish *Service* requests per *Session* shall be configurable. | |

### 5.3 Other Features

Table 12 describes Base features related items that can be profiled. For additional information about these items, please refer to Part 6.

**Table 12 – Base Information**

| Category | Title | Description | Derived |
|---|---|---|---|
| *Server* | Base Info Core Structure | Support *Server Object*, *Server* Capabilities; support the OPC UA *AddressSpace* structure. | |
| *Server* | Base Info Diagnostics | Support Diagnostic *Objects* and *Variables*. | |
| *Server* | Base Info Type System | Present a Type System with Data Types, *Reference* Types, *Object Types* and *Variable* Types. This includes custom types for any non-standard types | |
| *Server* | Base Info Model Change | Support ModelChange *Event* and NodeVersion *Property* for some *Nodes*. | |
| *Client* | Base Info *Client* Basic | Use OPC UA defined *AddressSpace* structure. Access or provide access to *Server* information like the *Server*'s state, BuildInfo, capabilities, Namespace Table and Type Model. | |
| *Client* | Base Info *Client* Type Programming | Programmatically process instances of *Objects* or *Variables* by using their type definitions. This includes custom Data Types, *Object Types* and *Variable* Types | |
| *Client* | Base Info *Client* Change Events | Process ModelChange Events to detect changes in the *Server*'s *AddressSpace* and take appropriate action. | |
| *Client* | Base Info *Client* RefreshData Mathod | Use the RefreshData *Method* to receive all current samples of monitored items. | |

Table 13 describes Security related units that can be profiled. 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. .

**Table 13 – Security**

| Category | Title | Description | Derived |
|---|---|---|---|
| Security | Security User Name Password | User name password combination. Specific encryption of the password is required if no Message encryption is used. | |

| Category | Title | Description | Derived |
|---|---|---|---|
| Security | Security User X509 | A public/private key pair. Must be able to be adminstrativly disabled | |
| Security | Security *Certificate* Validation | *Certificate* will be validated as specified in OPC UA 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<br>SymmetricSignatureAlgorithm -<br>SymmetricEncryptionAlgorithm -<br>AsymmetricSignatureAlgorithm -<br>SymmetricKeyWrapAlgorithm -<br>AsymmetricEncryptionAlgorithm -<br>KeyDerivationAlgorithm PSha1<br>(http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/dk/p_sha1)<br>DerivedSignatureKeyLength 0 | |
| Security | Security None CreateSession ActivateSession | The CreateSession and ActivateSession service allow for a NULL/empty signiture and do not require Application *Certificates* or a Nonce. This Conformance Unit is only valid for select security policies. | |
| Security | Security Basic 128Rsa15 | A suite of algorithms that uses RSA15 as Key-Wrap-algorithm and 128-Bit for encryption algorithms.<br>SymmetricSignatureAlgorithm HmacSha1<br>(http://www.w3.org/2000/09/xmldsig#hmac-sha1)<br>SymmetricEncryptionAlgorithm Aes128<br>(http://www.w3.org/2001/04/xmlenc#aes128-cbc)<br>AsymmetricSignatureAlgorithm RsaSha1<br>(http://www.w3.org/2000/09/xmldsig#rsa-sha1)<br>AsymmetricKeyWrapAlgorithm KwRsa15<br>(http://www.w3.org/2001/04/xmlenc#rsa-1_5)<br>AsymmetricEncryptionAlgorithm Rsa15<br>(http://www.w3.org/2001/04/xmlenc#rsa-1_5)<br>KeyDerivationAlgorithm PSha1<br>(http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/dk/p_sha1)<br>DerivedSignatureKeyLength 128 | |

| Category | Title | Description | Derived |
|---|---|---|---|
| Security | Security Basic 256 | A suite of algorithms that 256-Bit for encryption algorithms<br>SymmetricSignatureAlgorithm HmacSha1 (http://www.w3.org/2000/09/xmldsig#hmac-sha1)<br>SymmetricEncryptionAlgorithm Aes256 (http://www.w3.org/2001/04/xmlenc#aes256-cbc)<br>AsymmetricSignatureAlgorithm RsaSha1 (http://www.w3.org/2000/09/xmldsig#rsa-sha1)<br>AsymmetricKeyWrapAlgorithm KwRsaOaep (http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p)<br>AsymmetricEncryptionAlgorithm RsaOaep (http://www.w3.org/2001/04/xmlenc#rsa-oaep)<br>KeyDerivationAlgorithm PSha1 (http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/dk/p_sha1)<br>DerivedSignatureKeyLength 192 | |
| Security | Security Administration | Allow configuration of the following Security related items.<br>* select the proper User identification policy (User Name/Password or X509 or Kerberos)<br>* enable/disable the Security Policy "None" or other policies<br>* set the permitted certification authorities<br>* define how to react to unknown *Certificates* | |
| Security | Security Administration - XML Schema | Support the OPC defined xml schema for importing and export security configuration information. | |
| Security | Security *Certificate* Administration | Allow the plant administrator to assign a plant specific instance *Certificate* and allow configuration of a plant specific CA for acceptance of *Certificates* | |

Table 14 describes Protocol and encoding related features that can be profiled. These features are defined in detail in Part 6. It is recommended that a *Server* and *Client* support as many of these options as possible for greatest interoperability.

**Table 14 – Protocol and Encoding**

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

Table 15 describes Address Space Model information related items that can be profiled. The details of these model items is defined in Part 3 and Part 5

**Table 15 – Address Space Model**

| Category | Title | Description | Derived |
|---|---|---|---|
| *Server* | Address Space Base | Support the *NodeClasses* with their *Attributes* and behaviour as defined in OPC UA Part 3. This includes for instance: *Object*, *Object Type*, *Variable*, *Variable* Type, *References* and Data Type | |
| *Server* | Address Space Complex Datatypes | Support ComplexData with Data Dictionary. | |
| *Server* | Address Space *Method* | Support *Method Nodes*. | |
| *Client* | Address Space *Client* Base | Use and understand the *NodeClasses* with their *Attributes* and behaviour as defined in OPC UA Part 3. This includes for instance: *Object*, *Object Type*, *Variable*, *Variable* Type, *References* and Data Type | |

Table 16 describes Data Access information model related items that can be profiled. The details of this model are defined in Part 8.

**Table 16 – Data Access**

| Category | Title | Description | Derived |
|---|---|---|---|
| *Server* | Data Access DataItems | Provide *Variables* of DataItemType or one of its subtypes. Support the StatusCodes specified in the DataAccess part of the OPC UA specification. Support of optional Properties (e.g. "InstrumentRange") shall be verified during certification testing and will be shown in the *Certificate*. | |
| *Server* | Data Access Analog | Support AnalogItemType *Variables* with corresponding Properties. | |
| *Server* | Data Access PercentDeadBand | Support PercentDeadband filter when monitoring AnalogItemType *Variables*. | |
| *Server* | Data Access Semantic Changes | Support semantic changes of AnalogItemType items (EURange *Property* and/or EngineeringUnits *Property*). Support semantic change StatusCode bits where appropriate. | |
| *Server* | Data Access TwoState | Support TwoStateDiscreteType *Variables* with corresponding Properties. | |
| *Server* | Data Access MultiState | Support MultiStateDiscreteType *Variables* with corresponding Properties. | |
| *Client* | Data Access *Client* Basic | Understand the DataAccess *Variable* Types. Make use of the standard Properties if applicable. | |
| *Client* | Data Access *Client* Deadband | Use PercentDeadband to filter value changes of AnalogItemType *Variables*. | |
| *Client* | Data Access *Client* SemanticChange | Recognize the semantic change bit in the StatusCode while monitoring items and take proper action. Typically, the *Client* has to re-read Properties that define type-specific semantic like the EURange and EngineeringUnit Properties. | |

Table 17 describes *Alarm* and *Conditions* information model related items that can be profiled. The details of this model are defined in Part 9

**Table 17 – Alarm & Conditions**

| Category | Title | Description | Derived |
|---|---|---|---|
| | Undefined | No Conformance Units defined | |

Table 18 describes Historical Data Access information model related items that can be profiled. The details of this model are defined in Part 11.

**Table 18 – Historical Access**

| Category | Title | Description | Derived |
|---|---|---|---|
| | Undefined | No Conformance Units defined | |

Table 19 describes Command information model related items that can be profiled.

**Table 19 – Programs**

| Category | Title | Description | Derived |
|---|---|---|---|
| | Undefined | No Conformance Units defined | |

Table 20 describes Auditing related items that can be profiled.

**Table 20 – Auditing**

| Category | Title | Description | Derived |
|---|---|---|---|
| *Server* | Auditing Base | Support AuditEvents. The list of supported AuditEvents shall be verified during certification testing and will be shown in the *Certificate*. | |
| *Client* | Auditing *Client* Audit ID | *Client* support generating AuditEvents ids and providing them to *Servers* | |
| *Client* | Auditing *Client* Subscribes | The *Client* supports subscribing for audit Events and storing /processing them in a secure manner. | |

Table 21 describes Redundancy related items that are profiled.

**Table 21 – Redundancy**

| Category | Title | Description | Derived |
|---|---|---|---|
| *Server* | Redundancy *Server* | Support *Server*-based redundancy. | |
| *Server* | Redundancy *Server* Transparent | Support transparent *Server* redundancy. | |
| *Client* | Redundancy *Client* | *Client* supports *Client* redundancy. *Clients* that support *Client* redundancy can fail over to another *Client* (requires some out of band communication) | |
| *Client* | Redundancy *Client* Switch | *Clients* supporting this *ConformanceUnit* monitor the redundancy status for non-transparent redundancy *Servers* and switch to the backup *Server* when they recognize a change. | |

# 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** 22 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 can be used to access the OPC Foundation WEB site and obtain additional information about the *Profile*. This URI is also included in the *SoftwareCertificate* associated with the *Profile*. The OPC Foundation WEB site may also contain additional categories, which are specific to OPC Foundation generated companion specifications, but not part of the OPC UA core specifications.

An application (*Client* or *Server*) shall implement all of the *ConformanceUnits* in a *Profile*, 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. For example, 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 support for optional *ConformanceUnits* is described in the certificate that is generated by the associated testing.

**Table 22 – Profile List**

| ***Profile*** | **Description** | **Related category** | **URI** |
|---|---|---|---|
| *AddressSpace* Lookup *Client* Facet | includes all *ConformanceUnits* needed to navigate through the *AddressSpace*. | *Client* | http://opcfoundation.org/UA-*Profile*/*Client*/AddressSpaceLookup |
| Advanced Type Programming *Client* Facet | specifies the behaviour of *Clients* when processing instances based on Type Definitions. | *Client* | http://opcfoundation.org/UA-*Profile*/*Client*/TypeProgramming |
| *Attribute* Read *Client* Facet | includes *ConformanceUnits* needed to read *Attribute* values of *Nodes*. | *Client* | http://opcfoundation.org/UA-*Profile*/*Client*/AttributeRead |
| *Attribute* Write *Client* Facet | includes *ConformanceUnits* needed to write *Attribute* values of *Nodes*. | *Client* | http://opcfoundation.org/UA-*Profile*/*Client*/AttributeWrite |
| Auditing *Client* Facet | includes *ConformanceUnits* for the purpose of Auditing. | *Client* | http://opcfoundation.org/UA-*Profile*/*Client*/Auditing |
| Base *Client* Behaviour Facet | specifies behavioural aspects that *Clients* shall follow for best use by operators and administrators. These aspects can only be tested in a test lab. They are best practice guidelines. | *Client* | http://opcfoundation.org/UA-*Profile*/*Client*/Behaviour |

| ***Profile*** | **Description** | **Related category** | **URI** |
|---|---|---|---|
| Core *Client* Facet | defines core functionality for *Clients*. This facet includes the core functions for Security and *Session* handling. | *Client* | http://opcfoundation.org/UA-*Profile*/*Client*/Core |
| DataAccess *Client* Facet | defines *ConformanceUnits* needed for utilizing the DataAccess Information Model. | *Client* | http://opcfoundation.org/UA-*Profile*/*Client*/DataAccess |
| DataChange Subscriber *Client* Facet | includes *ConformanceUnits* needed when monitoring *Attribute* values for data change. | *Client* | http://opcfoundation.org/UA-*Profile*/*Client*/DataChangeSubscriber |
| *Discovery Client* Facet | includes *ConformanceUnits* for discovery of *Servers* and their Endpoints. | *Client* | http://opcfoundation.org/UA-*Profile*/*Client*/*Discovery* |
| *Event* Subscriber *Client* Facet | includes *ConformanceUnits* needed when subscribing for *Event Notifications*. | *Client* | http://opcfoundation.org/UA-*Profile*/*Client*/EventSubscriber |
| *Method Client* Facet | includes *ConformanceUnits* needed to call Methods. | *Client* | http://opcfoundation.org/UA-*Profile*/*Client*/*Method* |
| *Node* Management *Client* Facet | includes the *ConformanceUnits* that *Clients* will have to use to configure the *AddressSpace* of an OPC UA *Server* through OPC UA *Node* Management *Service* Set. | *Client* | http://opcfoundation.org/UA-*Profile*/*Client*/NodeManagement |
| Redundancy Switch *Client* Facet | specifies the ability to monitor the redundancy status in non-transparent redundancy *Servers* and switch to the backup *Server* when they recognize a change in status. | *Client* | http://opcfoundation.org/UA-*Profile*/*Client*/RedundancySwitch |
| Redundant *Client* Facet | defines *ConformanceUnits* that redundant *Clients* have to provide. | *Client* | http://opcfoundation.org/UA-*Profile*/*Client*/Redundancy |
| SecurityPolicy - Basic128Rsa15 | defines a Security Policy for medium to highly secure configurations. | Security | http://opcfoundation.org/UA/SecurityPolicy#Basic128Rsa15 |
| SecurityPolicy - Basic256 | defines a Security Policy for configurations with high security needs. | Security | http://opcfoundation.org/UA/SecurityPolicy/Basic256 |
| SecurityPolicy - None | defines a policy used for configurations with lowest security needs. This Security Policy can affect the behaviour of the CreateSession and Activate *Session* services. It also results in a SecureChannel which has no Channel Security | Security | http://opcfoundation.org/UA/SecurityPolicy#None |
| Auditing *Server* Facet | specifies Auditing functionality for *Servers*. | *Server* | http://opcfoundation.org/UA-*Profile*/*Server*/Auditing |
| Base *Server* Behaviour Facet | specifies behavioural aspects that *Servers* shall follow for best use by operators and administrators. These aspects can only be tested in a test lab. They are best practice guidelines. | *Server* | http://opcfoundation.org/UA-*Profile*/*Server*/Behaviour |
| Basic DataChange *Subscription Server* Facet | includes the *ConformanceUnits* when supporting basic DataChange *Subscription* functionality. | *Server* | http://opcfoundation.org/UA-*Profile*/*Server*/BaseDataChangeSubscription |

| ***Profile*** | **Description** | **Related category** | **URI** |
|---|---|---|---|
| *Client* Redundancy Facet | includes the *ConformanceUnits* to support redundant *Clients*. | *Server* | http://opcfoundation.org/UA-*Profile*/*Server*/ClientRedundancy |
| ComplexType *Server* Facet | defines a set of *ConformanceUnits* that define *Server* specific handling of Complex Types. | *Server* | http://opcfoundation.org/UA-*Profile*/*Server*/ComplexTypes |
| Core *Server* Facet | defines core functionality for all *Servers*.<br>This facet identifies *Discovery*, *Session*, *AddressSpace* and Viewing capabilities. | *Server* | http://opcfoundation.org/UA-*Profile*/*Server*/CoreFacet |
| DataAccess *Server* Facet | includes *ConformanceUnits* that define the handling of the DataAccess Information Model. | *Server* | http://opcfoundation.org/UA-*Profile*/*Server*/DataAccess |
| Embedded UA *Server* | focus is for devices with more than 50 Megabyte memory and with CPUs comparable to Intel 486 processor. | *Server* | http://opcfoundation.org/UA-*Profile*/*Server*/EmbeddedUA |
| Enhanced DataChange *Subscription Server* Facet | defines the typical DataChange *Subscription* functionality to be provided by OPC UA *Servers*. | *Server* | http://opcfoundation.org/UA-*Profile*/*Server*/EnhancedDataChangeSubscription |
| *Event Subscription Server* Facet | defines the functionality when supporting *Subscription* functionality for Events. | *Server* | http://opcfoundation.org/UA-*Profile*/*Server*/EventSubscription |
| Low End Embedded Device *Server* | defines a *Profile* which is appropriate only for small devices with limited resources, including the inability to include a security infrastructure. It is expected that most applications will use gateways that enhance the functionality of these *Servers* to Embedded or Standard UA *Server Profiles*. | *Server* | http://opcfoundation.org/UA-*Profile*/*Server*/LowEndEmbeddedDevice |
| *Method Server* Facet | specifies the *Method* handling functionality for *Servers*. | *Server* | http://opcfoundation.org/UA-*Profile*/*Server*/Methods |
| *Node* Management *Server* Facet | includes the *ConformanceUnits* when supporting configuration of the *AddressSpace* through the OPC UA *Node* Management *Service* Set. | *Server* | http://opcfoundation.org/UA-*Profile*/*Server*/NodeManagement |
| Redundancy Transparent *Server* Facet | includes the *ConformanceUnits* for *Servers* with transparent redundancy. | *Server* | http://opcfoundation.org/UA-*Profile*/*Server*/TransparentRedundancy |
| Redundancy Visible *Server* Facet | includes the *ConformanceUnits* for visible redundancy. | *Server* | http://opcfoundation.org/UA-*Profile*/*Server*/VisibleRedundancy |
| Standard UA *Server* | defines the most common functionality for OPC UA *Servers*. | *Server* | http://opcfoundation.org/UA-*Profile*/*Server*/StandardUA |
| SOAP-HTTP WS-SC UA Binary | defines a Stack *Profile* which represents the combination of HTTP/SOAP 1.2, WS-SecureConversation 1.3 and UA Binary 1.0 encoding. | Transport | http://opcfoundation.org/UA-*Profile*/Transport/soaphttp-wssc-uabinary |

| ***Profile*** | **Description** | **Related category** | **URI** |
|---|---|---|---|
| SOAP-HTTP WS-SC UA XML | defines a Stack *Profile* which represents the combination of HTTP/SOAP 1.2, WS-SecureConversation 1.3 and UA XML 1.0 encoding. | Transport | http://opcfoundation.org/UA-*Profile*/Transport/soaphttp-wssc-uaxml |
| SOAP-HTTP WS-SC UA XML-UA Binary | defines a Stack *Profile* represents the combination of HTTP/SOAP 1.2, WS-SecureConversation 1.3 and UA XML 1.0 and UA Binary 1.0 encodings. | Transport | http://opcfoundation.org/UA-*Profile*/Transport/soaphttp-wssc-uaxml-uabinary |
| UA-TCP UA-SC UA Binary | defines a Stack *Profile* which represents the combination of UA TCP 1.0, UA SecureConversation 1.0 and UA Binary 1.0 encoding. | Transport | http://opcfoundation.org/UA-*Profile*/Transport/uatcp-uasc-uabinary |

The contents of each of the listed *Profiles* will be described in 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* are composed of *ConformanceUnits*. More than one *Profile* may contain the same *ConformanceUnit*. *Profiles* may recursively contain other *Profiles*

*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, will review the list of available profiles. From this list the vendor will 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” *Profile*, the “UA-TCP UA-SC UA Binary” *Profile*, the “Data Access *Client* Facet” *Profile*, the “DataChange Subscriber *Client* Facet“ *Profile* and the “*Attribute* Write *Client* Facet” *Profile*. If the *Client* is to be lab 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 data, write 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 just an illustration and the represented *Profiles* may change.

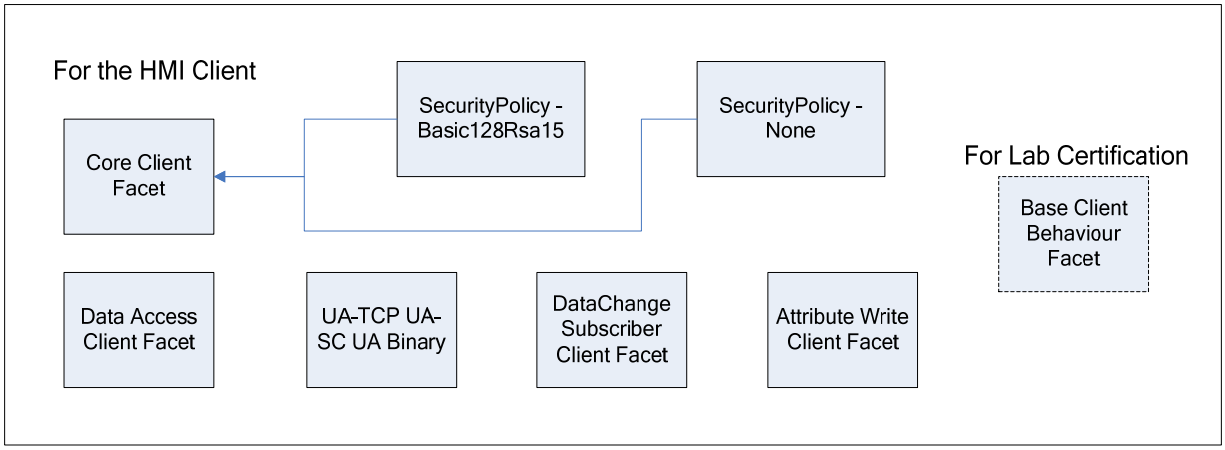


**Figure 2 - HMI Client Sample**

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 low-end 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.

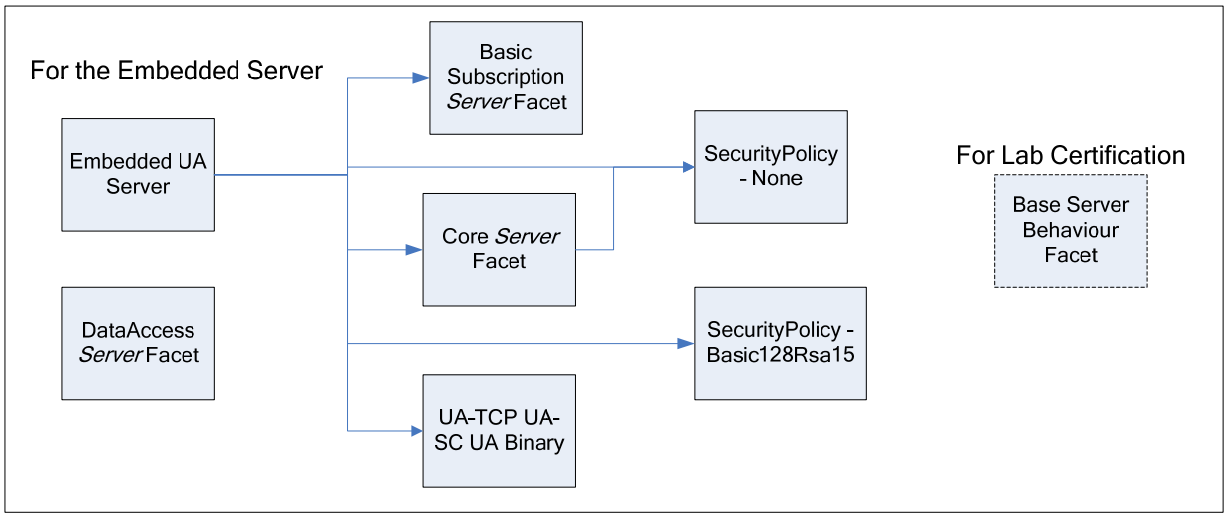


**Figure 3 - Embedded Server Sample**

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.

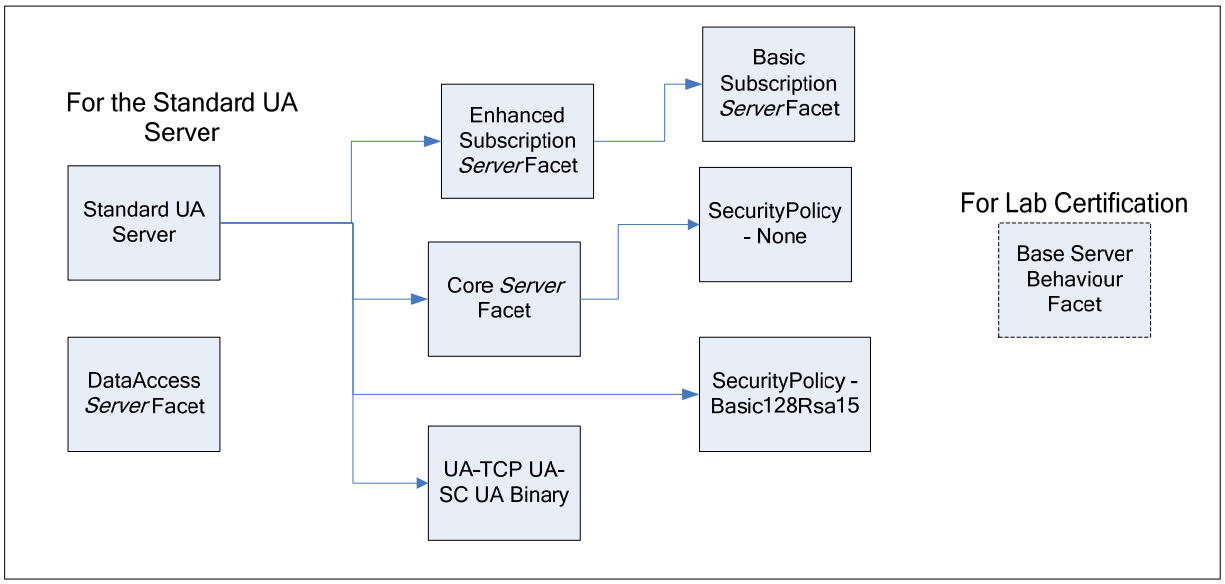


**Figure 4 - Standard UA Server Sample**

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 subscription or sessions based on the performance limits of the server. If the HMI client 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.

If an End User were examining OPC UA *Servers* or *Clients* for purchase the user would examine the *Profiles* supported by each product. They would also check that the *Profiles* that describe the functionality they desire are available from both the OPC UA *Server* and OPC UA *Client* that they wish to purchase. The *Profile* and *ConformanceUnit* descriptions are available via the OPC Foundation WEB site. The *Profiles* supported by OPC UA *Clients* and *Servers* and the level of testing that the OPC UA Applications have undergone are also available on the OPC Foundation WEB site. End Users are encouraged to use the OPC Foundation WEB Site to review OPC UA products before purchasing them.

## 6.5 Profile Tables

### 6.5.1 Introduction

The following sections describe the *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 AddressSpace Lookup Client Facet

Table 23 describes the details of the *AddressSpace* Lookup *Client* Facet *Profile*. This *Profile* includes all *ConformanceUnits* needed to navigate through the *AddressSpace*.

.

**Table 23 – 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 |
| Base Information | Base Info *Client* Basic | False |
| Base Information | Base Info *Client* Change Events | 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 |

### 6.5.3 Advanced Type Programming Client Facet

Table 24 describes the details of the Advanced Type Programming *Client* Facet *Profile*. This *Profile* specifies the behaviour of *Clients* when processing instances based on Type Definitions.

.

**Table 24 – 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 |

### 6.5.4 Attribute Read Client Facet

Table 25 describes the details of the *Attribute* Read *Client* Facet *Profile*. This *Profile* includes *ConformanceUnits* needed to read *Attribute* values of *Nodes*.

.

**Table 25 – 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 |

### 6.5.5 Attribute Write Client Facet

Table 26 describes the details of the *Attribute* Write *Client* Facet *Profile*. This *Profile* includes *ConformanceUnits* needed to write *Attribute* values of *Nodes*.

.

**Table 26 – 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 |

### 6.5.6 Auditing Client Facet

Table 27 describes the details of the Auditing *Client* Facet *Profile*. This *Profile* includes *ConformanceUnits* for the purpose of Auditing.

.

**Table 27 – 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 |

### 6.5.7 Base Client Behaviour Facet

Table 28 describes the details of the Base *Client* Behaviour Facet *Profile*. This *Profile* specifies behavioural aspects that *Clients* shall follow for best use by operators and administrators. These aspects can only be tested in a test lab. They are best practice guidelines.

.

**Table 28 – 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 |

### 6.5.8 Core Client Facet

Table 29 describes the details of the Core *Client* Facet *Profile*. This *Profile* defines core functionality for *Clients*. This facet includes the core functions for Security and *Session* handling.

.

**Table 29 – Core Client Facet**

| Group | Conformance Unit / *Profile* Title | Optional |
|---|---|---|
| *Profile* | SecurityPolicy - Basic128Rsa15 | False |
| *Profile* | SecurityPolicy - None | False |
| Security | Security Administration | False |
| Security | Security User Name Password | False |
| Security | Security User X509 | 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 |

### 6.5.9 DataAccess Client Facet

Table 30 describes the details of the DataAccess *Client* Facet *Profile*. This *Profile* defines *ConformanceUnits* needed for utilizing the DataAccess Information Model.

.

**Table 30 – DataAccess 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* Read Complex | False |

| Group | Conformance Unit / *Profile* Title | Optional |
|---|---|---|
| *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 |

### 6.5.10 DataChange Subscriber Client Facet

Table 31 describes the details of the DataChange Subscriber *Client* Facet *Profile*. This *Profile* includes *ConformanceUnits* needed when monitoring *Attribute* values for data change.

.

**Table 31 – DataChange Subscriber Client Facet**

| Group | Conformance Unit / *Profile* Title | Optional |
|---|---|---|
| Monitored Item *Services* | Monitor *Client* by Index | False |
| 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 |

### 6.5.11 Discovery Client Facet

Table 32 describes the details of the *Discovery Client* Facet *Profile*. This *Profile* includes *ConformanceUnits* for discovery of *Servers* and their Endpoints.

.

**Table 32 – 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 |

### 6.5.12 Event Subscriber Client Facet

Table 33 describes the details of the *Event* Subscriber *Client* Facet *Profile*. This *Profile* includes *ConformanceUnits* needed when subscribing for *Event Notifications*.

.

**Table 33 – Event Subscriber Client Facet**

| Group | Conformance Unit / *Profile* Title | Optional |
|---|---|---|
| Address Space Model | Address Space *Client* Base | False |
| Monitored Item *Services* | Monitor *Client* Complex Evt 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 |
| *Subscription Services* | *Subscription Client* Basic | False |
| *Subscription Services* | *Subscription Client* Modify | True |
| *Subscription Services* | *Subscription Client* Multiple | True |
| *Subscription Services* | *Subscription Client* Republish | False |

| Group | Conformance Unit / *Profile* Title | Optional |
|---|---|---|
| View *Services* | View *Client* Basic Browse | False |
| View *Services* | View *Client* TranslateBrowsePath | False |

### 6.5.13 Method Client Facet

Table 34 describes the details of the *Method Client* Facet *Profile*. This *Profile* includes *ConformanceUnits* needed to call Methods.

.

**Table 34 – Method Client Facet**

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

### 6.5.14 Node Management Client Facet

Table 35 describes the details of the *Node* Management *Client* Facet *Profile*. This *Profile* includes the *ConformanceUnits* that *Clients* will have to use to configure the *AddressSpace* of an OPC UA *Server* through OPC UA *Node* Management *Service* Set.

.

**Table 35 – 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 |

### 6.5.15 Redundancy Switch Client Facet

Table 36 describes the details of the Redundancy Switch *Client* Facet *Profile*. This *Profile* specifies the ability to monitor the redundancy status in non-transparent redundancy *Servers* and switch to the backup *Server* when they recognize a change in status.

.

**Table 36 – Redundancy Switch Client Facet**

| Group | Conformance Unit / *Profile* Title | Optional |
|---|---|---|
| Base Information | Base Info *Client* RefreshData Mathod | True |
| Redundancy | Redundancy *Client* Switch | False |

### 6.5.16 Redundant Client Facet

Table 37 describes the details of the Redundant *Client* Facet *Profile*. This *Profile* defines *ConformanceUnits* that redundant *Clients* have to provide.

.

**Table 37 – Redundant Client Facet**

| Group | Conformance Unit / *Profile* Title | Optional |
|---|---|---|
| Base Information | Base Info *Client* RefreshData Mathod | True |
| Redundancy | Redundancy *Client* | False |

### 6.5.17 SecurityPolicy - Basic128Rsa15

Table 38 describes the details of the SecurityPolicy - Basic128Rsa15 *Profile*. This *Profile* defines a Security Policy for medium to highly secure configurations.

.

**Table 38 – SecurityPolicy - Basic128Rsa15**

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

### 6.5.18 SecurityPolicy - Basic256

Table 39 describes the details of the SecurityPolicy - Basic256 *Profile*. This *Profile* defines a Security Policy for configurations with high security needs.

.

**Table 39 – SecurityPolicy - Basic256**

| Group | Conformance Unit / *Profile* Title | Optional |
|---|---|---|
| Security | Security Basic 256 | False |
| Security | Security *Certificate* Validation | False |

### 6.5.19 SecurityPolicy - None

Table 40 describes the details of the SecurityPolicy - None *Profile*. This *Profile* defines a policy used for configurations with lowest security needs. This Security Policy can affect the behaviour of the CreateSession and Activate *Session* services. It also results in a SecureChannel which has no Channel Security

.

**Table 40 – SecurityPolicy - None**

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

### 6.5.20 Auditing Server Facet

Table 41 describes the details of the Auditing *Server* Facet *Profile*. This *Profile* specifies Auditing functionality for *Servers*.

.

**Table 41 – Auditing Server Facet**

| Group | Conformance Unit / *Profile* Title | Optional |
|---|---|---|
| Auditing | Auditing Base | False |

### 6.5.21 Base Server Behaviour Facet

Table 42 describes the details of the Base *Server* Behaviour Facet *Profile*. This *Profile* specifies behavioural aspects that *Servers* shall follow for best use by operators and administrators. These aspects can only be tested in a test lab. They are best practice guidelines.

.

**Table 42 – 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 |

### 6.5.22 Basic DataChange Subscription Server Facet

Table 43 describes the details of the Basic DataChange *Subscription Server* Facet *Profile*. This *Profile* includes the *ConformanceUnits* when supporting basic DataChange *Subscription* functionality.

.

**Table 43 – Basic DataChange Subscription Server Facet**

| Group | Conformance Unit / *Profile* Title | Optional |
|---|---|---|
| Monitored Item *Services* | Monitor Basic | False |
| Monitored Item *Services* | Monitor Items 100 | False |
| Monitored Item *Services* | Monitor QueueSize_1 | False |
| Monitored Item *Services* | Monitor Triggering | False |
| Monitored Item *Services* | Monitor Value Change | 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 |

### 6.5.23 Client Redundancy Facet

Table 44 describes the details of the *Client* Redundancy Facet *Profile*. This *Profile* includes the *ConformanceUnits* to support redundant *Clients*.

.

**Table 44 – Client Redundancy Facet**

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

### 6.5.24 ComplexType Server Facet

Table 45 describes the details of the ComplexType *Server* Facet *Profile*. This *Profile* defines a set of *ConformanceUnits* that define *Server* specific handling of Complex Types.

.

**Table 45 – 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 |

### 6.5.25 Core Server Facet

Table 46 describes the details of the Core *Server* Facet *Profile*. This *Profile* defines core functionality for all *Servers*.
This facet identifies *Discovery*, *Session*, *AddressSpace* and Viewing capabilities.

.

**Table 46 – Core Server Facet**

| Group | Conformance Unit / *Profile* Title | Optional |
|---|---|---|
| *Profile* | SecurityPolicy - None | False |
| Address Space Model | Address Space Base | False |
| *Attribute Services* | *Attribute* Read | False |

| Group | Conformance Unit / *Profile* Title | Optional |
|---|---|---|
| *Attribute Services* | *Attribute* Write Index | True |
| *Attribute Services* | *Attribute* Write Values | True |
| Base Information | Base Info Core Structure | False |
| *Discovery Services* | *Discovery* Find *Servers* Self | False |
| *Discovery Services* | *Discovery* Get Endpoints | False |
| Security | Security Administration | False |
| Security | Security User Name Password | False |
| Security | Security User X509 | False |
| *Session Services* | *Session* Base | False |
| *Session Services* | *Session* General *Service* Behaviour | False |
| *Session Services* | *Session* Minimum 10 Parallel | False |
| View *Services* | View Basic | False |
| View *Services* | View Minimum Continuation Point 01 | False |
| View *Services* | View RegisterNodes | False |
| View *Services* | View TranslateBrowsePath | False |

### 6.5.26 DataAccess Server Facet

Table 47 describes the details of the DataAccess *Server* Facet *Profile*. This *Profile* includes *ConformanceUnits* that define the handling of the DataAccess Information Model.

.

**Table 47 – DataAccess Server Facet**

| Group | Conformance Unit / *Profile* Title | Optional |
|---|---|---|
| Data Access | Data Access Analog | True |
| Data Access | Data Access DataItems | False |
| 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 |

### 6.5.27 Embedded UA Server

Table 48 describes the details of the Embedded UA *Server Profile*. This *Profile* focus is for devices with more than 50 Megabyte memory and with CPUs comparable to Intel 486 processor.

.

**Table 48 – Embedded UA Server**

| Group | Conformance Unit / *Profile* Title | Optional |
|---|---|---|
| *Profile* | Basic DataChange *Subscription Server* Facet | False |
| *Profile* | Low End Embedded Device *Server* | False |
| *Profile* | SecurityPolicy - Basic128Rsa15 | False |
| Base Information | Base Info Type System | False |

### 6.5.28 Enhanced DataChange Subscription Server Facet

Table 49 describes the details of the Enhanced DataChange *Subscription Server* Facet *Profile*. This *Profile* defines the typical DataChange *Subscription* functionality to be provided by OPC UA *Servers*.

.

**Table 49 – Enhanced DataChange Subscription Server Facet**

| Group | Conformance Unit / *Profile* Title | Optional |
|---|---|---|
| *Profile* | Basic DataChange *Subscription Server* Facet | False |
| Monitored Item *Services* | Monitor Items 500 | False |

| Group | Conformance Unit / *Profile* Title | Optional |
|---|---|---|
| Monitored Item *Services* | Monitor MinQueueSize_02 | False |
| *Subscription Services* | *Subscription* Minimum 05 | False |
| *Subscription Services* | *Subscription* Publish Min 10 | False |

### 6.5.29 Event Subscription Server Facet

Table 50 describes the details of the *Event Subscription Server* Facet *Profile*. This *Profile* defines the functionality when supporting *Subscription* functionality for Events.

.

**Table 50 – Event Subscription Server Facet**

| Group | Conformance Unit / *Profile* Title | Optional |
|---|---|---|
| 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_1 | 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 |

### 6.5.30 Low End Embedded Device Server

Table 51 describes the details of the Low End Embedded Device *Server Profile*. This *Profile* defines a *Profile* which is appropriate only for small devices with limited resources, including the inability to include a security infrastructure. It is expected that most applications will use gateways that enhance the functionality of these *Server*.....

**Table 51 – Low End Embedded Device Server**

| Group | Conformance Unit / *Profile* Title | Optional |
|---|---|---|
| *Profile* | Core *Server* Facet | False |
| *Profile* | UA-TCP UA-SC UA Binary | False |

### 6.5.31 Method Server Facet

Table 52 describes the details of the *Method Server* Facet *Profile*. This *Profile* specifies the *Method* handling functionality for *Servers*.

.

**Table 52 – Method Server Facet**

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

### 6.5.32 Node Management Server Facet

Table 53 describes the details of the *Node* Management *Server* Facet *Profile*. This *Profile* includes the *ConformanceUnits* when supporting configuration of the *AddressSpace* through the OPC UA *Node* Management *Service* Set.

.

**Table 53 – Node Management Server Facet**

| Group | Conformance Unit / *Profile* Title | Optional |
|---|---|---|

| 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 Redundancy Transparent Server Facet

Table 54 describes the details of the Redundancy Transparent *Server* Facet *Profile*. This *Profile* includes the *ConformanceUnits* for *Servers* with transparent redundancy.

.

**Table 54 – Redundancy Transparent Server Facet**

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

### 6.5.34 Redundancy Visible Server Facet

Table 55 describes the details of the Redundancy Visible *Server* Facet *Profile*. This *Profile* includes the *ConformanceUnits* for visible redundancy.

.

**Table 55 – Redundancy Visible Server Facet**

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

### 6.5.35 Standard UA Server

Table 56 describes the details of the Standard UA *Server Profile*. This *Profile* defines the most common functionality for OPC UA *Servers*.

.

**Table 56 – Standard UA Server**

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

### 6.5.36 SOAP-HTTP WS-SC UA Binary

Table 57 describes the details of the SOAP-HTTP WS-SC UA Binary *Profile*. This *Profile* defines a Stack *Profile* which represents the combination of HTTP/SOAP 1.2, WS-SecureConversation 1.3 and UA Binary 1.0 encoding.

.

**Table 57 – SOAP-HTTP WS-SC UA Binary**

| Group | Conformance Unit / *Profile* Title | Optional |
|---|---|---|
| Protocol and Encoding | Protocol Soap Binary WS Security | False |

### 6.5.37 SOAP-HTTP WS-SC UA XML

Table 58 describes the details of the SOAP-HTTP WS-SC UA XML *Profile*. This *Profile* defines a Stack *Profile* which represents the combination of HTTP/SOAP 1.2, WS-SecureConversation 1.3 and UA XML 1.0 encoding.

.

**Table 58 – SOAP-HTTP WS-SC UA XML**

| Group | Conformance Unit / *Profile* Title | Optional |
|---|---|---|
| Protocol and Encoding | Protocol Soap Xml WS Security | False |

### 6.5.38 SOAP-HTTP WS-SC UA XML-UA Binary

Table 59 describes the details of the SOAP-HTTP WS-SC UA XML-UA Binary *Profile*. This *Profile* defines a Stack *Profile* represents the combination of HTTP/SOAP 1.2, WS-SecureConversation 1.3 and UA XML 1.0 and UA Binary 1.0 encodings.

.

**Table 59 – SOAP-HTTP WS-SC UA XML-UA Binary**

| Group | Conformance Unit / *Profile* Title | Optional |
|---|---|---|
| Protocol and Encoding | Protocol Soap Binary WS Security | False |
| Protocol and Encoding | Protocol Soap Xml WS Security | False |

### 6.5.39 UA-TCP UA-SC UA Binary

Table 60 describes the details of the UA-TCP UA-SC UA Binary *Profile*. This *Profile* defines a Stack *Profile* which represents the combination of UA TCP 1.0, UA SecureConversation 1.0 and UA Binary 1.0 encoding.

.

**Table 60 – UA-TCP UA-SC UA Binary**

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