

BS EN 62541-7:2012



BSI Standards Publication

OPC unified architecture

Part 7: Profiles

NO COPYING WITHOUT BSI PERMISSION EXCEPT AS PERMITTED BY COPYRIGHT LAW

raising standards worldwide™

Full document details: BS EN 62541-7:2012



National foreword

This British Standard is the UK implementation of EN 62541-7:2012. It is identical to IEC 62541-7:2012.

The UK participation in its preparation was entrusted to Technical Committee AMT/7, Industrial communications: process measurement and control, including fieldbus.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2012

Published by BSI Standards Limited 2012

ISBN 978 0 580 65224 0

ICS 25.040.40; 25.100.01

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 October 2012.

Amendments issued since publication

Amd. No.	Date	Text affected
----------	------	---------------

EUROPEAN STANDARD
NORME EUROPÉENNE
EUROPÄISCHE NORM

EN 62541-7

September 2012

ICS 25.040.40; 25.100.01

English version

OPC unified architecture - Part 7: Profiles (IEC 62541-7:2012)

Architecture unifiée OPC -
Partie 7: Profils
(CEI 62541-7:2012)

OPC Unified Architecture -
Teil 7: Profile
(IEC 62541-7:2012)

This European Standard was approved by CENELEC on 2012-09-04. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

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

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

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

CENELEC

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

Management Centre: Avenue Marnix 17, B - 1000 Brussels

© 2012 CENELEC - All rights of exploitation in any form and by any means reserved worldwide for CENELEC members.

Ref. No. EN 62541-7:2012 E

 **Documents**
Document Sharing Hub

Foreword

The text of document 65E/242/FDIS, future edition 1 of IEC 62541-7, prepared by SC 65E "Devices and integration in enterprise systems" of IEC/TC 65 "Industrial-process measurement, control and automation" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN 62541-7:2012.

The following dates are fixed:

- latest date by which the document has to be implemented at national level by publication of an identical national standard or by endorsement (dop) 2013-06-04
- latest date by which the national standards conflicting with the document have to be withdrawn (dow) 2015-09-04

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

Endorsement notice

The text of the International Standard IEC 62541-7:2012 was approved by CENELEC as a European Standard without any modification.

Annex ZA (normative)

Normative references to international publications with their corresponding European publications

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.

NOTE When an international publication has been modified by common modifications, indicated by (mod), the relevant EN/HD applies.

<u>Publication</u>	<u>Year</u>	<u>Title</u>	<u>EN/HD</u>	<u>Year</u>
IEC/TR 62541-1	-	OPC unified architecture - Part 1: Overview and concepts	CLC/TR 62541-1	-
IEC/TR 62541-2	-	OPC unified architecture - Part 2: Security model	CLC/TR 62541-2	-
IEC 62541-3	-	OPC unified architecture - Part 3: Address space model	EN 62541-3	-
IEC 62541-4	-	OPC unified architecture - Part 4: Services	EN 62541-4	-
IEC 62541-5	-	OPC unified architecture - Part 5: Information model	EN 62541-5	-
IEC 62541-6	-	OPC unified architecture - Part 6: Mappings	EN 62541-6	-
IEC 62541-8	-	OPC unified architecture - Part 8: Data access	EN 62541-8	-
IEC 62541-9	-	OPC unified architecture - Part 9: Alarms and conditions	EN 62541-9	-

CONTENTS

INTRODUCTION	9
1 Scope	10
2 Normative references	10
3 Terms, definitions and abbreviations	11
3.1 Terms and definitions	11
3.2 Abbreviations	11
4 Overview	12
4.1 General	12
4.2 ConformanceUnit	13
4.3 Profiles	13
4.4 Profile categories	14
5 ConformanceUnits	14
5.1 Overview	14
5.2 Services	15
5.3 Other Features	23
6 Profiles	29
6.1 Overview	29
6.2 Profile List	30
6.3 Conventions for Profile definitions	32
6.4 Applications	32
6.5 Profile Tables	34
6.5.1 General	34
6.5.2 A & C Address Space Instance Client Facet	34
6.5.3 A & C Alarm Client Facet	35
6.5.4 A & C Dialog Client Facet	36
6.5.5 A & C Enable Client Facet	36
6.5.6 A & C Exclusive Alarming Client Facet	37
6.5.7 A & C Non-Exclusive Alarming Client Facet	37
6.5.8 A & C Previous Instances Client Facet	38
6.5.9 A & C Simple Client Facet	39
6.5.10 A & E Proxy Facet	39
6.5.11 AddressSpace Lookup Client Facet	40
6.5.12 Advanced Type Programming Client Facet	41
6.5.13 Attribute Read Client Facet	41
6.5.14 Attribute Write Client Facet	42
6.5.15 Auditing Client Facet	42
6.5.16 Base Client Behaviour Facet	43
6.5.17 Core Client Facet	44
6.5.18 DataAccess Client Facet	44
6.5.19 DataChange Subscriber Client Facet	45
6.5.20 Discovery Client Facet	45
6.5.21 Event Subscriber Client Facet	46
6.5.22 Method Client Facet	47
6.5.23 Node Management Client Facet	47

6.5.24	Redundancy Switch Client Facet	47
6.5.25	Redundant Client Facet	48
6.5.26	SecurityPolicy – Basic128Rsa15.....	48
6.5.27	SecurityPolicy – Basic256	48
6.5.28	SecurityPolicy – None.....	49
6.5.29	A & C Acknowledgeable Alarm Server Facet.....	49
6.5.30	A & C Address Space Instance Server Facet	49
6.5.31	A & C Alarm Server Facet.....	49
6.5.32	A & C Dialog Server Facet.....	50
6.5.33	A & C Enable Server Facet.....	50
6.5.34	A & C Exclusive Alarming Server Facet	50
6.5.35	A & C Non-Exclusive Alarming Server Facet.....	51
6.5.36	A & C Previous Instances Server Facet	51
6.5.37	A & C Simple Server Facet	51
6.5.38	A & E Wrapper Facet.....	52
6.5.39	Address Space Notifier Server Facet	52
6.5.40	Auditing Server Facet	53
6.5.41	Base Server Behaviour Facet	53
6.5.42	Client Redundancy Facet.....	53
6.5.43	ComplexType Server Facet	54
6.5.44	Core Server Facet	54
6.5.45	Data Access Server Facet	55
6.5.46	Embedded DataChange Subscription Server Facet.....	55
6.5.47	Embedded UA Server Profile	55
6.5.48	Enhanced DataChange Subscription Server Facet.....	56
6.5.49	Method Server Facet	56
6.5.50	Micro Embedded Device Server	56
6.5.51	Nano Embedded Device Server	57
6.5.52	Node Management Server Facet	57
6.5.53	Redundancy Transparent Server Facet.....	57
6.5.54	Redundancy Visible Server Facet	58
6.5.55	Standard DataChange Subscription Server Facet	58
6.5.56	Standard Event Subscription Server Facet.....	58
6.5.57	Standard UA Server.....	59
6.5.58	SOAP-HTTP WS-SC UA Binary	59
6.5.59	SOAP-HTTP WS-SC UA XML	60
6.5.60	SOAP-HTTP WS-SC UA XML-UA Binary	60
6.5.61	UA-TCP UA-SC UA Binary.....	60
	Bibliography.....	61
	Figure 1 – Profile – ConformanceUnit – TestCases	13
	Figure 2 – HMI Client Sample	33
	Figure 3 – Embedded Server Sample	33
	Figure 4 – Standard UA Server Sample	34
	Table 1 – Profile Categories	14
	Table 2 – Conformance Groups	15

Table 3 – Discovery Services	16
Table 4 – Session Services	17
Table 5 – Node Management Services	18
Table 6 – View Services	18
Table 7 – Query Services	19
Table 8 – Attribute Services	19
Table 9 – Method Services	20
Table 10 – Monitored Item Services	20
Table 11 – Subscription Services	22
Table 12 – Base Information	23
Table 13 – Security	24
Table 14 – Protocol and Encoding	26
Table 15 – Address Space Model	26
Table 16 – Data Access	26
Table 17 – Alarms and Conditions	27
Table 18 – Auditing	29
Table 19 – Redundancy	29
Table 20 – Profile List	30
Table 21 – A & C Address Space Instance Client Facet	35
Table 22 – A & C Address Space Instance Client Facet Related Server Profiles	35
Table 23 – A & C Alarm Client Facet	35
Table 24 – A & C Alarm Client Facet Related Server Profiles	35
Table 25 – A & C Dialog Client Facet	36
Table 26 – A & C Dialog Client Facet Related Server Profiles	36
Table 27 – A & C Enable Client Facet	36
Table 28 – A & C Enable Client Facet Related Server Profiles	37
Table 29 – A & C Exclusive Alarming Client Facet	37
Table 30 – A & C Exclusive Alarming Client Facet Related Server Profiles	37
Table 31 – A & C Non-Exclusive Alarming Client Facet	38
Table 32 – A & C Non-Exclusive Alarming Client Facet Related Server Profiles	38
Table 33 – A & C Previous Instances Client Facet	38
Table 34 – A & C Previous Instances Client Facet Related Server Profiles	38
Table 35 – A & C Simple Client Facet	39
Table 36 – A & C Simple Client Facet Related Server Profiles	39
Table 37 – A & E Proxy Facet	39
Table 38 – AddressSpace Lookup Client Facet	40
Table 39 – AddressSpace Lookup Client Facet Related Server Profiles	41
Table 40 – Advanced Type Programming Client Facet	41
Table 41 – Advanced Type Programming Client Facet Related Server Profiles	41
Table 42 – Attribute Read Client Facet	42
Table 43 – Attribute Read Client Facet Related Server Profiles	42
Table 44 – Attribute Write Client Facet	42
Table 45 – Attribute Write Client Facet Related Server Profiles	42

Table 46 – Auditing Client Facet	43
Table 47 – Auditing Client Facet Related Server Profiles	43
Table 48 – Base Client Behaviour Facet	43
Table 49 – Base Client Behaviour Facet Related Server Profiles	43
Table 50 – Core Client Facet	44
Table 51 – Core Client Facet Related Server Profiles	44
Table 52 – DataAccess Client Facet	44
Table 53 – DataAccess Client Facet Related Server Profiles	45
Table 54 – DataChange Subscriber Client Facet	45
Table 55 – DataChange Subscriber Client Facet Related Server Profiles	45
Table 56 – Discovery Client Facet	45
Table 57 – Discovery Client Facet Related Server Profiles	46
Table 58 – Event Subscriber Client Facet	46
Table 59 – Event Subscriber Client Facet Related Server Profiles	46
Table 60 – Method Client Facet	47
Table 61 – Method Client Facet Related Server Profiles	47
Table 62 – Node Management Client Facet	47
Table 63 – Node Management Client Facet Related Server Profiles	47
Table 64 – Redundancy Switch Client Facet	48
Table 65 – Redundancy Switch Client Facet Related Server Profiles	48
Table 66 – Redundant Client Facet	48
Table 67 – Redundant Client Facet Related Server Profiles	48
Table 68 – SecurityPolicy – Basic128Rsa15	48
Table 69 – SecurityPolicy – Basic256	49
Table 70 – SecurityPolicy – None	49
Table 71 – A & C Acknowledgeable Alarm Server Facet	49
Table 72 – A & C Address Space Instance Server Facet	49
Table 73 – A & C Alarm Server Facet	50
Table 74 – A & C Dialog Server Facet	50
Table 75 – A & C Enable Server Facet	50
Table 76 – A & C Exclusive Alarming Server Facet	51
Table 77 – A & C Non-Exclusive Alarming Server Facet	51
Table 78 – A & C Previous Instances Server Facet	51
Table 79 – A & C Simple Server Facet	52
Table 80 – A & E Wrapper Facet	52
Table 81 – Address Space Notifier Server Facet	53
Table 82 – Auditing Server Facet	53
Table 83 – Base Server Behaviour Facet	53
Table 84 – Client Redundancy Facet	53
Table 85 – ComplexType Server Facet	54
Table 86 – Core Server Facet	54
Table 87 – Data Access Server Facet	55
Table 88 – Embedded DataChange Subscription Server Facet	55

Table 89 – Embedded UA Server Profile	56
Table 90 – Enhanced DataChange Subscription Server Facet.....	56
Table 91 – Method Server Facet	56
Table 92 – Micro Embedded Device Server.....	57
Table 93 – Nano Embedded Device Server.....	57
Table 94 – Node Management Server Facet	57
Table 95 – Redundancy Transparent Server Facet.....	57
Table 96 – Redundancy Visible Server Facet.....	58
Table 97 – Standard DataChange Subscription Server Facet	58
Table 98 – Standard Event Subscription Server Facet	59
Table 99 – Standard UA Server	59
Table 100 – SOAP-HTTP WS-SC UA Binary	60
Table 101 – SOAP-HTTP WS-SC UA XML	60
Table 102 – SOAP-HTTP WS-SC UA XML-UA Binary	60
Table 103 – UA-TCP UA-SC UA Binary.....	60

INTRODUCTION

This International Standard is a specification intended 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 inter-operate seamlessly together.

OPC UNIFIED ARCHITECTURE –

Part 7: Profiles

1 Scope

This part of the IEC 62541 series describes the OPC Unified Architecture *Profiles*. The *Profiles* 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 parts of IEC 62541.

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*.

Profiles are used to segregate features with regard to testing of OPC UA Products and the nature of the testing. This includes the testing performed by the OPC Foundation provided OPC UA Compliance 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 verse lab based testing. The scope of this specification 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.

2 Normative references

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

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

IEC/TR 62541-2, *OPC Unified Architecture – Part 2: Security Model*

IEC 62541-3, *OPC unified architecture – Part 3: Address Space Model*

IEC 62541-4, *OPC unified architecture – Part 4: Services*

IEC 62541-5, *OPC unified architecture – Part 5: Information Model*

IEC 62541-6, *OPC unified architecture – Part 6: Mappings*

IEC 62541-8, *OPC unified architecture – Part 8: Data Access*

IEC 62541-9, *OPC Unified architecture – Part 9: Alarms and conditions*

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 62541-1, IEC 62541-2, IEC 62541-3, IEC 62541-4, IEC 62541-6, IEC 62541-8 and the following apply.

3.1.1

Application

a body of code or piece of hardware 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

specific set of 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 section 4.2

3.1.3

ConformanceGroup

group of conformance units 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 specifications.

3.1.4

Facet

profile that is a subset of the *ConformanceUnits* a *Server* or *Client* may require

Note 1 to entry: *Facet* are typically combined to form full function *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

ProfileCategory

a grouping of *Profiles* into major functionality groups, such as *Server* or *Clients*

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

3.1.6

TestCase

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. A *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.

3.1.7

TestLab

location that is designed 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 specification describing OPC UA TestLabs and the testing they are to provide (see *Compliance Part 8 UA Server*, *Compliance Part 9 UA Client*).

3.2 Abbreviations

A&C Alarms and Conditions

A&E	Alarms and Events
API	Application Programming Interface
DA	Data Access
HA	Historical Access
HMI	Human Machine Interface
UA	Unified Architecture

4 Overview

4.1 General

The OPC Unified Architecture 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 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 *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, 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 specification. This relationship is also displayed by the OPC UA Compliance Test Tool.

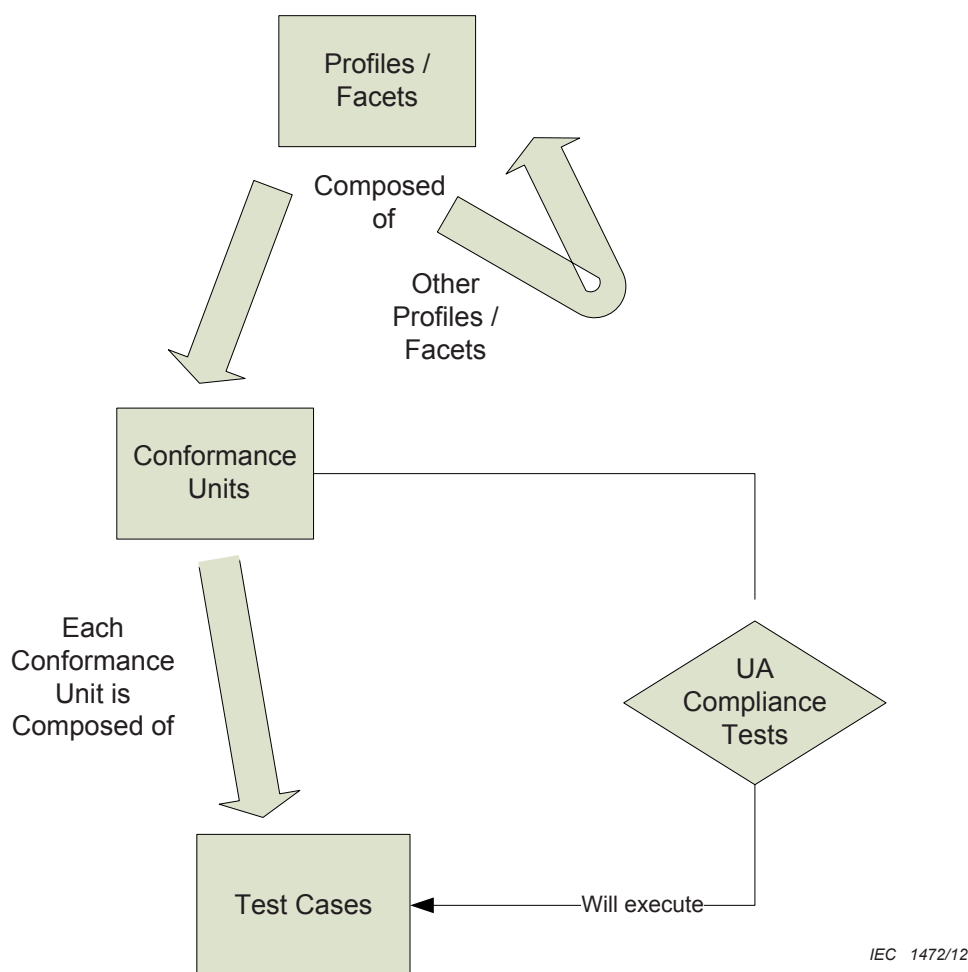


Figure 1 – Profile – ConformanceUnit – TestCases

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 the OPC UA specification. Additional information regarding testing of a *ConformanceUnit* is provided in the *Compliance Part 8 UA Server* or *Compliance Part 9 UA Client* test specifications.

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 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*. Categories include: *Server* and *Client*, but other example could also include power generation or chemical plant. A *Profile* shall be assigned to one or more categories.

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

Table 1 – Profile Categories

Category	Description
<i>Client</i>	<i>Profiles</i> of this category specify a complete functional set for an OPC UA <i>Client</i> . The URI of such profiles has to be part of a Software <i>Certificate</i> passed in the CreateSession request.
<i>Discovery Server</i>	<i>Profiles</i> of this category are for <i>Discovery Servers</i>
External	<i>Profiles</i> that are defined outside
Security	<i>Profiles</i> 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.
<i>Server</i>	<i>Profiles</i> of this category specify a complete functional set for an OPC UA <i>Server</i> . The URI of such profiles has to be part of a Software <i>Certificate</i> returned with the CreateSession service response.
Transport	<i>Profiles</i> 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 *ConformanceGroups*. These groups closely correspond to the *Service Sets* in IEC 62541-4 and to the OPC UA information models. Table 2 provides a list of the *ConformanceGroups*. These groups and the *ConformanceUnits* that they describe are detailed in the following subclauses. *ConformanceGroups* have no impact on testing; they are used only for organizational reasons, i.e. to simplify the readability of this document.

Table 2 – Conformance Groups

Group	Description
Address Space Model	Defines <i>ConformanceUnits</i> for various features of the OPC UA <i>AddressSpace</i> .
Aggregates	All conformance units that are related to Aggregates, including individual conformance units for each supported aggregate.
<i>Alarms and Conditions</i>	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> .
<i>Attribute Services</i>	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 IEC 62541-5 of the OPC UA specification.
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 IEC 62541-8 of the OPC UA specification.
<i>Discovery Services</i>	<i>ConformanceUnits</i> with focus on <i>Server</i> Endpoint <i>Discovery</i> .
<i>Method Services</i>	Methods represent the function calls of <i>Objects</i> . Methods are invoked and return only after completion (successful or unsuccessful).
Monitored Item <i>Services</i>	<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> .
<i>Node Management Services</i>	Bundles <i>ConformanceUnits</i> for all <i>Services</i> to add and delete OPC UA <i>AddressSpace</i> <i>Nodes</i> and <i>References</i> .
Protocol and Encoding	Covers all transport and encoding combinations that are specified in IEC 62541-6 of the OPC UA specification.
<i>Query Services</i>	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 <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.
<i>Session Services</i>	An (OPC UA) <i>Session</i> is an application layer connection.
<i>Subscription Services</i>	Subscriptions are used to report <i>Notifications</i> to the <i>Client</i> .
View <i>Services</i>	<i>Clients</i> use the View <i>Service</i> Set 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

The following tables describe *ConformanceUnits* for the *Services* specified in IEC 62541-4. 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* 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. 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	Derived
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 securityMode other than "None".	
Server	Discovery Configuration	Allow configuration of the <i>Discovery Server</i> URL where the <i>Server</i> will register itself. Allow complete disabling of registration with a <i>Discovery Server</i> .	
Client	Discovery Client Find Servers Basic	Uses the FindServers <i>Service</i> to obtain all <i>Servers</i> installed on a given platform.	
Client	Discovery Client Find Servers with URI	Use FindServers <i>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 FindServers <i>Service</i> call.	
Client	Discovery Client Get Endpoints Basic	Uses the GetEndpoints <i>Service</i> to obtain all Endpoints for a given <i>Server</i> URI.	
Client	Discovery Client Get Endpoints Dynamic	Detect changes to the Endpoints after an initial GetEndpoints <i>Service</i> call.	
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	Derived
Server	Session General Service Behaviour	Implement basic <i>Service</i> behaviour. This includes in particular: – 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</i> Set (CreateSession, ActivateSession, CloseSession) except the use of ActivateSession to change the <i>Session</i> user. This includes correct handling of all parameters that are provided. Note that for the CreateSession and ActivationSession services, if the SecurityMode = None then: 1) The Application <i>Certificate</i> and Nonce are optional. 2) The signatures are null/empty.	
Server	Session Cancel	Support the Cancel <i>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 Sessions (total for all <i>Clients</i>).	Session Minimum 1
Server	Session Minimum 50 Parallel	Support minimum 50 parallel Sessions (total for all <i>Clients</i>).	Session Minimum 2 Parallel
Client	Session Client General Service Behaviour	Implement basic <i>Service</i> behaviour. This includes in particular: – including the proper authentication token of the <i>Session</i> – creating requestHandles if needed – requesting diagnostic information with the 'returnDiagnostics' parameter – evaluate the serviceResult and operational results	
Client	Session Client Base	Use the <i>Session Service</i> Set (CreateSession, ActivateSession, CloseSession) except the use of ActivateSession to change the <i>Session</i> user. This includes correct handling of all parameters that are provided Note that for the CreateSession and ActivationSession services, if the SecurityMode = None then: 1) The Application <i>Certificate</i> and Nonce are optional. 2) The signatures are null/empty.	
Client	Session Client Renew NodeIds	This <i>ConformanceUnit</i> applies to <i>Clients</i> that persist NodeIds. 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	Use of ActivateSession 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 ServerStatus/State <i>Variable</i> to recognize a potential shutdown of the <i>Server</i> and clean up resources.	

Category	Title	Description	Derived
<i>Client</i>	<i>Session Client Cancel</i>	Use the <i>Cancel Service</i> to cancel outstanding requests.	
<i>Client</i>	<i>Session Client Auto Reconnect</i>	Automatic <i>Client</i> reconnect including: – <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	

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
<i>Server</i>	<i>Node Management Add Node</i>	Support the <i>AddNodes service</i> to add one or more <i>Nodes</i> into the OPC UA <i>AddressSpace</i> .	
<i>Server</i>	<i>Node Management Delete Node</i>	Support the <i>DeleteNodes Service</i> to delete one or more <i>Nodes</i> from the OPC UA <i>AddressSpace</i> .	
<i>Server</i>	<i>Node Management Add Ref</i>	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> .	
<i>Server</i>	<i>Node Management Delete Ref</i>	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> .	
<i>Client</i>	<i>Node Management Client</i>	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 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	Derived
<i>Server</i>	<i>View Basic</i>	Support the <i>View Service Set</i> (<i>Browse</i> , <i>BrowseNext</i>).	
<i>Server</i>	<i>View TranslateBrowsePath</i>	Support <i>TranslateBrowsePathsToNodeIds Service</i> .	
<i>Server</i>	<i>View RegisterNodes</i>	Support the <i>RegisterNodes</i> and <i>UnregisterNodes 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> .	
<i>Server</i>	<i>View Minimum Continuation Point 01</i>	Support minimum 1 continuation point per <i>Session</i> .	
<i>Server</i>	<i>View Minimum Continuation Point 05</i>	Support minimum 5 continuation points per <i>Session</i> . This number has to be supported for at least half of the minimum required sessions.	<i>View Minimum Continuation Point 01</i>
<i>Client</i>	<i>View Client Basic Browse</i>	Use <i>Browse</i> and <i>BrowseNext Services</i> to navigate through the <i>Server's</i> OPC UA <i>AddressSpace</i> . Make use of the <i>referenceTypeId</i> and the <i>nodeClassMask</i> to specify the needed <i>References</i> .	
<i>Client</i>	<i>View Client Basic ResultSet Filtering</i>	Make use of the <i>resultMask</i> parameter to optimize the result set to be returned by the <i>Server</i> .	

Category	Title	Description	Derived
<i>Client</i>	View <i>Client</i> TranslateBrowsePath	Use the TranslateBrowsePathsToNodeIds <i>Service</i> to identify the NodeIds for <i>Nodes</i> where a starting <i>Node</i> and a BrowsePath is known. Make bulk operations rather than multiple calls whenever possible.	
<i>Client</i>	View <i>Client</i> RegisterNodes	Use 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.	

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 and as such is supported by most *Servers*. Most *Clients* will also support some aspects of the *Attribute Service* Set

Table 8 – Attribute Services

Category	Title	Description	Derived
<i>Server</i>	<i>Attribute</i> Read	Support 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	
<i>Server</i>	<i>Attribute</i> Read Complex	Support reading and encoding ComplexData (structures).	
<i>Server</i>	<i>Attribute</i> Write Values	Support writing to values to one or more <i>Attributes</i> of one or more <i>Nodes</i> .	
<i>Server</i>	<i>Attribute</i> Write Complex	Support writing and decoding ComplexData.	
<i>Server</i>	<i>Attribute</i> Write StatusCode & TimeStamp	Support writing of StatusCode and Timestamps along with the Value.	
<i>Server</i>	<i>Attribute</i> Write Index	Support the IndexRange to write a single element or a range of elements when the <i>Attribute</i> value is an array.	
<i>Server</i>	<i>Attribute</i> Alternate Encoding	Support 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 Stack <i>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.	
<i>Client</i>	<i>Attribute Client</i> 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.	
<i>Client</i>	<i>Attribute Client</i> 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> .	