

OPC 10000-19

OPC Unified Architecture Part 19: Dictionary Reference

Release 1.04

2020-03-05

Specification Type:	Industry Standard Specification	Comments:	Report or view errata: http://www.opcfoundation.org/errata
Document Number	OPC 10000-19		
Title:	OPC Unified Architecture Part 19 :Dictionary Reference	Date:	2020-03-05
Version:	<u>Release 1.04</u>	Software:	MS-Word
		Source:	OPC 10000-19 - UA Specification Part 19 - Dictionary Reference 1.04.docx
Author:	<u>OPC Foundation</u>	Status:	Release

CONTENTS

		Page
1	Scope	1
2	Normative references	1
3	Terms, definitions, abbreviated terms, and conventions	1
	3.1 Terms and definitions	1
4	Dictionary Reference Information Model overview	1
	4.1 General	1
	4.2 Overview	2
5	OPC UA ObjectTypes	3
	5.1 DictionaryEntryType	3
	5.2 DictionaryFolderType	3
	5.3 IRDI ISO/IEC 11179-6 conformant DictionaryEntryType	3
	5.4 URI based dictionary entry type	3
6	OPC UA ReferenceTypes	3
	6.1 HasDictionaryEntry ReferenceType	3
7	OPC UA VariableTypes	4
	7.1 MultiStateDictionaryEntryDiscreteBaseType VariableType	4
	7.2 MultiStateDictionaryEntryDiscreteType VariableType	4
8	OPC UA Objects	5
	8.1 Dictionaries Object	5

FIGURES

Figure 1 – The dictionary reference types..... 3

TABLES

Table 1 – MultiStateDictionaryEntryDiscreteBaseType Definition 4
Table 2 – MultiStateDictionaryEntryDiscreteType Definition 4

OPC FOUNDATION

UNIFIED ARCHITECTURE

FOREWORD

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

Copyright © 2006-2020, OPC Foundation, Inc.

AGREEMENT OF USE

COPYRIGHT RESTRICTIONS

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

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

PATENTS

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

WARRANTY AND LIABILITY DISCLAIMERS

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

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

RESTRICTED RIGHTS LEGEND

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

COMPLIANCE

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

TRADEMARKS

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

GENERAL PROVISIONS

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

This Agreement shall be governed by and construed under the laws of the State of Minnesota, excluding its choice 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 19: Dictionary Reference

1 Scope

This specification defines an Information Model of the OPC Unified Architecture. The Information Model describes the basic infrastructure to reference from an OPC UA Information Model to external dictionaries like IEC Common Data Dictionary or eCI@ss.

2 Normative references

The following referenced documents are indispensable for the application of this specification. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments and errata) applies.

OPC 10000-1, *OPC Unified Architecture - Part 1: Overview and Concepts*
<http://www.opcfoundation.org/UA/Part1/>

OPC 10000-3, *OPC Unified Architecture - Part 3: Address Space Model*
<http://www.opcfoundation.org/UA/Part3/>

OPC 10000-4, *OPC Unified Architecture - Part 4: Services*
<http://www.opcfoundation.org/UA/Part4/>

OPC 10000-5, *OPC Unified Architecture - Part 5: Information Model*
<http://www.opcfoundation.org/UA/Part5/>

OPC 10000-8, *OPC Unified Architecture - Part 8: Data Access*
<http://www.opcfoundation.org/UA/Part8/>

OPC 10001-5, *OPC Unified Architecture V1.04 - Amendment 5: Dictionary Reference*
<http://www.opcfoundation.org/UA/Amendment5/>

3 Terms, definitions, abbreviated terms, and conventions

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in OPC 10000-1, OPC 10000-3, and OPC 10000-5 apply.

All used terms are *italicized* in the specification.

4 Dictionary Reference Information Model overview

4.1 General

This document describes the basic infrastructure an OPC UA Information Model may use to reference external dictionaries like IEC Common Data Dictionary or eCI@ss. It defines *ObjectTypes*, *VariableTypes* and a *ReferenceType* and explains how they should be used.

The *ObjectTypes* are used to represent an external dictionary in an OPC UA *AddressSpace*. The *ReferenceType* is used to reference from *Nodes* in the *AddressSpace* to the dictionary entries. Such dictionary entries can be seen as external classification or external semantic information.

The type system of OPC UA already provides means to express the semantic of an OPC UA Object. As an example, OPC 10000-100 defines the *DeviceType* *ObjectType* expressing that instances of this *ObjectType* represent devices. Subtypes of the *DeviceType* are used to add vendor specific semantic.

However, the classification and additional semantic of the device in terms of an external data dictionary is not specified further. This document provides means to represent that an Object is for example a differential pressure transmitter in the context of an IEC Common Data Dictionary. This allows clients to automatically retrieve and identify such devices.

This document is an integral part of this standard, that is, the types defined in this document shall be used as defined. However, it is not required but strongly recommended that a Server use the types defined in this document to refer to external dictionaries. The defined types may be subtyped to refine their behaviour.

When a Server references external dictionaries using the types defined in this document, it refers from OPC UA Nodes to dictionary entries. The Server may optionally also provide the hierarchy and content of the external dictionary. Resource consumption needs to be considered, especially in scenarios where the OPC UA Server is part of the firmware of a device.

Note: This version of the document reference Types defined in Amendment OPC 10001-5 and described as amendments to OPC 10001-5. In a future version of this document the Types defined in the Amendment will be included here and not in OPC 10001-5.

4.2 Overview

The types and instances defined in this document are illustrated in Figure 1. The *DictionaryEntryType* is an abstract base type for dictionary entries. The *IrdiDictionaryEntryType* and the *UriDictionaryEntryType* provide concrete types that can be used to represent dictionary entries. The *HasDictionaryEntry ReferenceType* is used to refer an OPC UA Node to a dictionary entry. The *DictionaryFolderType* and the *Dictionaries Object* provide an optional capability to expose the hierarchy of a referenced dictionary.

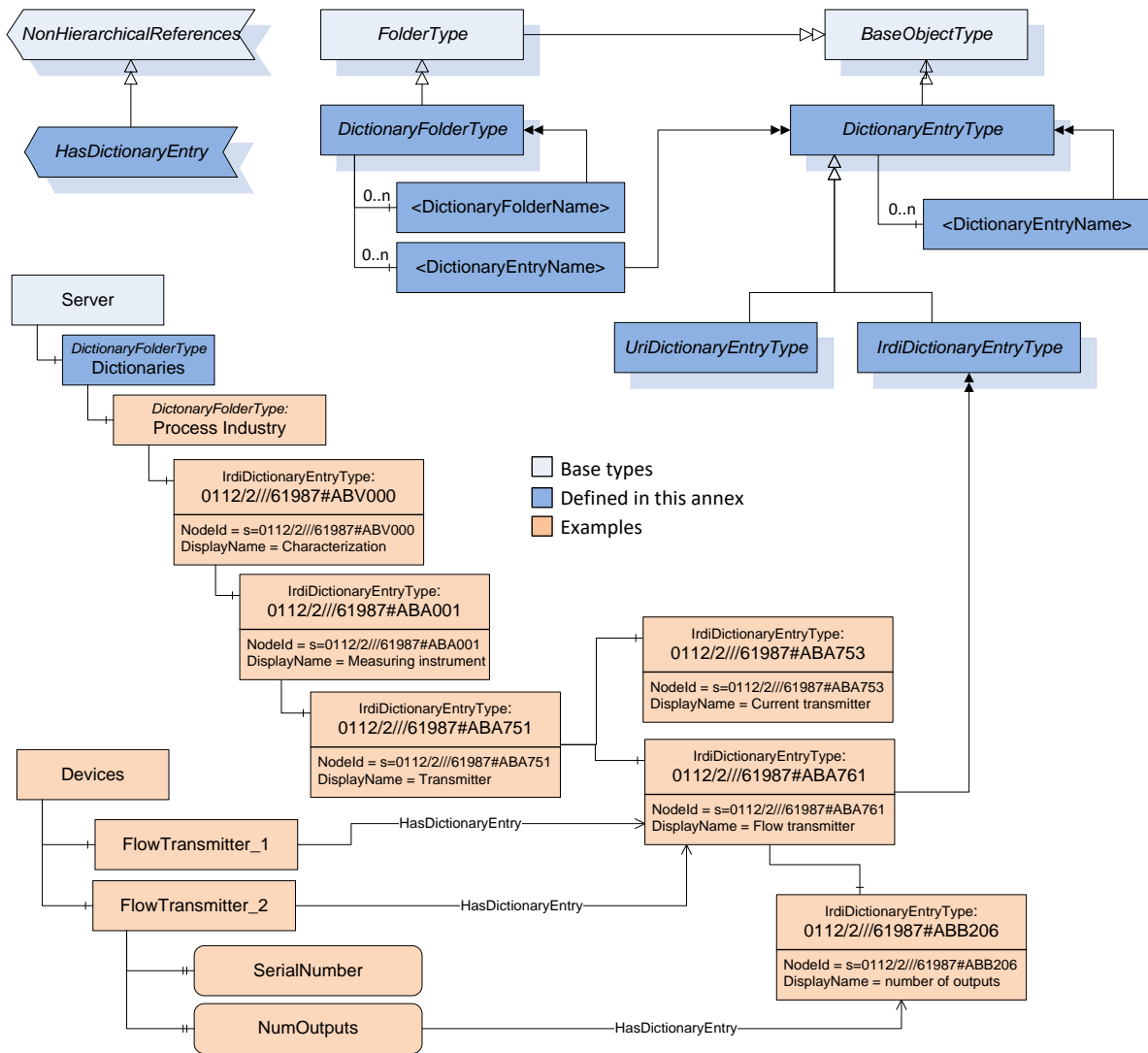


Figure 1 – The dictionary reference types

5 OPC UA ObjectTypes

5.1 DictionaryEntryType

This abstract *ObjectType* is formally defined in OPC 10001-5 for this version of the document.

5.2 DictionaryFolderType

This *ObjectType* is formally defined in OPC 10001-5 for this version of the document.

5.3 IRDI ISO/IEC 1179-6 conformant DictionaryEntryType

This *ObjectType* is formally defined in OPC 10001-5 for this version of the document.

5.4 URI based dictionary entry type

This *ObjectType* is formally defined in OPC 10001-5 for this version of the document.

6 OPC UA ReferenceTypes

6.1 HasDictionaryEntry ReferenceType

This *ReferenceType* is formally defined in OPC 10001-5 for this version of the document.

7 OPC UA VariableTypes

7.1 MultiStateDictionaryEntryDiscreteBaseType VariableType

The *MultiStateDictionaryEntryDiscreteBaseType VariableType* is a subtype of the *MultiStateValueDiscreteType*. It provides dictionary entries for each of the possible states as well as the current state of the *MultiStateValueDiscreteType*. It is formally defined in Table 1.

Table 1 – MultiStateDictionaryEntryDiscreteBaseType Definition

Attribute		Value			
BrowseName		MultiStateDictionaryEntryDiscreteBaseType			
IsAbstract		False			
ValueRank		Scalar			
DataType		Number			
References	NodeClass	BrowseName	DataType	TypeDefinition	Modelling Rule
Subtype of the MultiStateValueDiscreteType defined in OPC 10000-8					
HasProperty	Variable	EnumDictionaryEntries	NodeId[][]	PropertyType	Mandatory
HasComponent	Variable	ValueAsDictionaryEntries	NodeId[]	PropertyType	Optional

The normal approach of associating *DictionaryEntryType Nodes* with *HasDictionaryEntry References* cannot be used with a *MultiStateValueDiscreteType VariableType* due to the *EnumStrings Property* being an *Array*, making this *Type necessary*. *References* can however be used for other *VariableTypes*. For example, the *TwoStateDiscreteType* can use *HasDictionaryEntry References* with *SourceNode* being the *TrueState* and *FalseState Properties* and *TargetNode* being the *DictionaryEntry Nodes*.

EnumDictionaryEntries is a two-dimensional array of *NodeIds*. The first dimension is used to list all possible dictionary entry values for the related variable in a specific dictionary (e.g. CDD or eCI@ss). The second dimension is used to reference this dictionary. The size of the first array dimension shall be the same size as the *EnumStrings Property*.

ValueAsDictionaryEntries provides a list of all dictionary entry values in the different dictionaries related to the current value of the variable. The order of the array entries shall be in the same order used by the *EnumDictionaryEntries Property*.

If an instance of this type is writeable and the optional *ValueAsDictionaryEntries Property* is implemented, it shall be writeable as well. *Clients* writing to the *ValueAsDictionaryEntries Property* shall use one of the *DictionaryEntryType NodeIds* defined by the *EnumDictionaryEntries Property*. This will have the same result as writing the value attribute, but the client does not require knowledge of the values.

The *NodeIds* represent the dictionary entries and can be generated with dictionary knowledge.

7.2 MultiStateDictionaryEntryDiscreteType VariableType

The *MultiStateDictionaryEntryDiscreteType VariableType* is a subtype of the *MultiStateDictionaryEntryDiscreteBaseType*. It requires the *ValueAsDictionaryEntries Property*. It is formally defined in Table 2.

Table 2 – MultiStateDictionaryEntryDiscreteType Definition

Attribute		Value			
BrowseName		MultiStateDictionaryEntryDiscreteType			
IsAbstract		False			
ValueRank		Scalar			
DataType		Number			
References	NodeClass	BrowseName	DataType	TypeDefinition	Modelling Rule
Subtype of the MultiStateDictionaryEntryDiscreteBaseType defined in 7.1					
HasComponent	Variable	ValueAsDictionaryEntries	NodeId[]	PropertyType	Mandatory

8 OPC UA Objects

8.1 Dictionaries Object

This *ObjectType* is formally defined in OPC 10001-5 for this version of the document.
