BS IEC 61671-2:2016

BSI Standards Publication

Standard for automatic test markup language (ATML) instrument description

National foreword

This British Standard is the UK implementation of IEC 61671-2:2016.

The UK participation in its preparation was entrusted to Technical Committee EPL/501, Electronic Assembly Technology.

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.

Published by BSI Standards Limited 2016

ISBN 978 0 580 92302 9
ICS 25.040; 35.060

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 30 April 2016.

Amendments/corrigenda issued since publication

Date	Text affected

BS IEC 61671-2:2016

IEC 61671-2

Edition 1.0 2016-04

INTERNATIONAL STANDARD

IEEE Std 1671.2™

Standard for automatic test markup language (ATML) instrument description

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 25.040; 35.060

ISBN 978-2-8322-3265-1

Warning! Make sure that you obtained this publication from an authorized distributor.

Contents

STANDARD FOR AUTOMATIC TEST MARKUP LANGUAGE (ATML) INSTRUMENT DESCRIPTION

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation.

   IEEE Standards documents are developed within IEEE Societies and Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. IEEE develops its standards through a consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of IEEE and serve without compensation. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards. Use of IEEE Standards documents is wholly voluntary. IEEE documents are made available for use subject to important notices and legal disclaimers (see http://standards.ieee.org/IPR/disclaimers.html for more information).

   IEC collaborates closely with IEEE in accordance with conditions determined by agreement between the two organizations.

2) The formal decisions of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees. The formal decisions of IEEE on technical matters, once consensus within IEEE Societies and Standards Coordinating Committees has been reached, is determined by a balanced ballot of materially interested parties who indicate interest in reviewing the proposed standard. Final approval of the IEEE standards document is given by the IEEE Standards Association (IEEE-SA) Standards Board.

3) IEC/IEEE Publications have the form of recommendations for international use and are accepted by IEC National Committees/IEEE Societies in that sense. While all reasonable efforts are made to ensure that the technical content of IEC/IEEE Publications is accurate, IEC or IEEE cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.

4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications (including IEC/IEEE Publications) transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC/IEEE Publication and the corresponding national or regional publication shall be clearly indicated in the latter.

5) IEC and IEEE do not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC and IEEE are not responsible for any services carried out by independent certification bodies.

6) All users should ensure that they have the latest edition of this publication.

7) No liability shall attach to IEC or IEEE or their directors, employees, servants or agents including individual experts and members of technical committees and IEC National Committees, or volunteers of IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board, for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC/IEEE Publication or any other IEC or IEEE Publications.

8) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

9) Attention is drawn to the possibility that implementation of this IEC/IEEE Publication may require use of material covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. IEC or IEEE shall not be held responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patent Claims or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility.

International Standard IEC 61671-2/IEEE Std 1671.2-2012 has been processed through IEC technical committee 91: Electronics assembly technology, under the IEC/IEEE Dual Logo Agreement.

The text of this standard is based on the following documents:

IEEE Std	FDIS	Report on voting
IEEE Std 1671.2-2012	91/1314/FDIS	91/1338/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

The IEC Technical Committee and IEEE Technical Committee have decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC web site under "http://webstore.iec.ch" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IEEE Std 1671.2™-2012
(Revision of
IEEE Std 1671.2-2008)

IEEE Standard for Automatic Test Markup Language (ATML) Instrument Description

Sponsor

IEEE Standards Coordinating Committee 20 on Test and Diagnosis for Electronic Systems

Approved 5 December 2012

IEEE-SA Standards Board

Abstract: An exchange format is specified in this standard, using extensible markup language (XML), for identifying instrumentation that may be integrated in an automatic test system (ATS) that is to be used to test and diagnose a unit under test (UUT).

Keywords: ATML instance document, ATS, automatic test equipment (ATE), Automatic Test Markup Language (ATML), automatic test system, IEEE 1671.2, instrument, instrumentation, synthetic instrument, XML schema

---

IEEE Introduction

This introduction is not part of IEEE Std 1671.2-2012, IEEE Standard for Automatic Test Markup Language (ATML) Instrument Description.

This child, or "dot," standard, also known as an ATML component standard, provides for the definition of the *InstrumentDescription* and *InstrumentInstance* XML schemas, and contains references to examples. These XML schemas and examples accompany this standard and provide for the identification and definition of an instrument.

XML schemas define the basic information required within any test application and provide a vehicle for formally defining the test environment by defining a class hierarchy corresponding to these basic information entities and providing several methods within each to enable basic operations to be performed on these entities. ATML component standards within the ATML framework define the particular requirements within the test environment.

The Synthetic Instrument Working Group (SIWG) was formed, at Department of Defense request, to define synthetic instrumentation and its attributes and develop a framework that balances user and supplier objectives, facilitates rapid technology advancements and adaptation throughout the test life cycle, and complements/supports other relevant test and measurement industry activities.

The goals or desired effects of the SIWG activities were to:

a) Reduce the total cost of ownership of the automatic test system (ATS).
b) Reduce time to develop and field new or upgraded ATSs.
c) Provide greater flexibility to the war fighter through U.S. and coalition partner's interoperable ATSs.
d) Reduce the ATS's logistics footprint.
e) Reduce the ATS's physical footprint.
f) Improve the quality of test.

The SIWG addressed the reductions from the test and measurement perspective. The SIWG efforts resulted in both the definition of synthetic instruments and the specifications of their respective attributes.

Synthetic instruments were originally part of the IEEE Std 1671.2™-2008[1] standard, as both an example of *InstrumentDescription* instances as well as to provide a definition of the necessary parameters/attributes to document a synthetic instrument as defined by the SIWG. These synthetic instrument definitions have now been incorporated into their own IEEE project (P1871.2), and therefore their associated Annexes have been removed from later revisions of IEEE Std 1671.2. The *InstrumentDescription* template instance example for synthetic instruments are still provided as downloads of this standard to ensure continuity and support for existing users.

Template instance documents are used by vendors developing/providing synthetic instruments as the basis for documenting the synthetic instrument. The template instance document provides examples for each instrument vendor to follow. These templates will not validate against the schemas documented within this standard until actual values for the specifications are incorporated into the SI-based *InstrumentDescription* instance document.

[1] IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08855, USA (http://standards.ieee.org/).

Notice to users

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

This document is copyrighted by the IEEE. It is made available for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making this document available for use and adoption by public authorities and private users, the IEEE does not waive any rights in copyright to this document.

Updating of IEEE documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect. In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE-SA Website at http://standards.ieee.org/index.html or contact the IEEE at the address listed previously. For more information about the IEEE Standards Association or the IEEE standards development process, visit IEEE-SA Website at http://standards.ieee.org/index.html.

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: http://standards.ieee.org/findstds/errata/index.html. Users are encouraged to check this URL for errata periodically.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE-SA Website at http://standards.ieee.org/about/sasb/patcom/patents.html. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Standard for Automatic Test Markup Language (ATML) Instrument Description

IMPORTANT NOTICE: IEEE Standards documents are not intended to ensure safety, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

This IEEE document is made available for use subject to important notices and legal disclaimers. These notices and disclaimers appear in all publications containing this document and may be found under the heading "Important Notice" or "Important Notices and Disclaimers Concerning IEEE Documents." They can also be obtained on request from IEEE or viewed at http://standards.ieee.org/IPR/disclaimers.html.

1. Overview

1.1 General

Automatic test markup language (ATML) is a collection of IEEE standards and associated extensible markup language (XML) schemas that allows automatic test system (ATS) and test information to be exchanged in a common format adhering to the XML standard.[1]

The ATML framework and the ATML family of standards have been developed and are maintained under the guidance of the Test Information Integration (TII) Subcommittee of IEEE Standards Coordinating Committee 20 (SCC20) to serve as a comprehensive environment for integrating design data, test strategies, test requirements, test procedures, test results management, and test system implementations, while allowing test program (TP), test asset interoperability, and unit under test (UUT) data to be interchanged between heterogeneous systems.

This standard (as well as the XML schemas and XML instance document examples[2] that accompany this standard) is intended to be used in identifying and documenting instrumentation which may be utilized

[1]This information is given for the convenience of users of this standard and does not constitute an endorsement by the IEEE of this consortium standard. Equivalent standards or products may be used if they can be shown to lead to the same results.
[2]The schemas and examples that accompany this standard are available at http://standards.ieee.org/downloads/1671/1671.2-2012/.

during the testing of a particular unit under test. This information includes the mechanical, electrical, and software interfaces of the instrument.

1.2 Application of this document's annexes

This document includes four annexes.

Annex A, Annex B, Annex C, and Annex D are informative, and thus are provided strictly as information, for users, implementers, and maintainers of this document.

1.3 Scope

This standard defines an exchange format, utilizing extensible markup language (XML), for both the static description of instrument models, and the specific description of instrument instance information.

1.4 Application

This standard provides for the specification and identification of instrumentation that will be used for the purposes of testing a UUT. The specification and identification consist of, but are not limited to: physical characteristics, power requirements, operational requirements, calibration requirements, factory defaults, configuration options, capabilities, and interfaces (both hardware [HW] and software [SW]). This collection of information represents an entire "data package" for either a class or type of instrument (as represented by the InstrumentDescription.xsd schema defined in Clause 4 or a specific instrument (as represented by the InstrumentInstance.xsd schema defined in Clause 5).

Identifying an instrument provides for the unambiguous specification of a particular instrument, which may be utilized in a bench-test scenario, a piece of manual test equipment, or within automatic test equipment (ATE). This unambiguous specification shall be readable by both humans and machines. Humans may use the specification to identify and assemble the instrument into their test application. Machines may use the specification to verify that the testing need can be accomplished by the instrument in place.

Synthetic instruments link a series of HW and SW components with standardized interfaces to generate signals or make measurements using algorithmic numeric processing techniques. The goal is to decrease the total cost of ownership of ATS, to lessen the time to develop and field new or upgraded ATS, to reduce the test system logistics footprint, and finally, to improve the quality of test.

The information contained in XML documents conforming to this standard will be useful to the following:

a) Test program set (TPS) developers

b) TPS maintainers

c) ATE system developers

d) ATE system maintainers

e) Developers of ATML-based tools and systems

f) Instrument manufactures

1.5 Conventions used within this document

1.5.1 General

In accordance with the *IEEE Standards Style Manual* [B6][3], any schema examples will be shown in `Courier` font. In cases where instance document examples are necessary to depict the use of a schema type or element, such examples will also be shown in `Courier` font. When the characters "..." appear in an example, it indicates that the example component is incomplete.

All simple types, complex types, attribute groups, and elements will be listed; explanatory information will be provided, along with examples, if additional clarification is needed. The explanatory information will include information on the intended use of the elements and/or attributes where the name of the entity does not clearly indicate its intended use. For elements derived from another source type (e.g., an abstract type), only attributes that extend the source type will be listed; details regarding the base type will be listed along with the base type.

When referring to an attribute of an XML element, the convention of *[element]@[attribute]* will be used. In cases where an attribute name is referred to with no associated element, the attribute name will be enclosed in single quotes. Element and type names will always be set in italics when appearing in text.

This standard uses the vocabulary and definitions of relevant IEEE standards. In case of conflict of definitions, except for those portions quoted from standards, the following precedence shall be observed: 1) Clause 3, 2) The *IEEE Standards Dictionary Online*.[B5]

1.5.2 Precedence

The *InstrumentDescription* schema (InstrumentDescription.xsd) element, child element, and annotation information shall take precedence over the descriptive information contained in Clause 4.

The *InstrumentDescription* schema and the material contained in Clause 4 shall take precedence over the *InstrumentDescription* instance document information represented in Clause 5 as well as the examples in Annex B.

The *InstrumentInstance* schema (InstrumentInstance.xsd) element, child element, and annotation information shall take precedence over the descriptive information contained in Clause 5.

The *InstrumentInstance* schema and the material contained in Clause 5 shall take precedence over the *InstrumentDescription* instance document information represented in Clause 5 as well as the examples in Annex B.

1.5.3 Word usage

In accordance with the *IEEE Standards Style Manual* [B6], the word *shall* is used to indicate mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals *is required to*). The use of the word *must* is used only to describe unavoidable situations. The use of the word *will* is only used in statements of fact.

The word *should* is used to indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others (*should* equals *is recommended that*).

---

[3] The numbers in brackets correspond to those of the bibliography in Annex D.

The word *may* is used to indicate a course of action permissible within the limits of the standard (*may* equals *is permitted to*).

The word *can* is used for statements of possibility and capability (*can* equals *is able to*).

2. Normative references

The following referenced document is indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

IEEE Std 1671™, IEEE Standard for Automatic Test Markup Language (ATML) for Exchanging Automatic Test Equipment and Test Information via XML.[4,5]

3. Definitions, abbreviations, and acronyms

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* [B5][6] should be consulted for terms not defined in this clause. In the event a term is explicitly redefined, or defined in more detail in an ATML component standard, the component standards definition shall be normative for that ATML component standard.

abstract type: A declared type that can be used to define other types through derivation. Only non-abstract types derived from the declared type can be used in instance documents. When such a type is used, it must be identified by the xsi:type attribute.

automatic test markup language (ATML) instance document: See: instance document.

dynamic current: The rated current capacity of a particular VersaModule Eurocard (VME) extensions for instrumentation (VXI) or Peripheral Component Interconnect (PCI®) extensions for instrumentation (PXI®) backplane voltage for the frequencies from 20 Hz to 1 GHz.[7]

element: A bounded component of the logical structure of an extensible markup language (XML) document that has a type and that may have XML attributes and content.[8]

entity: Something that has a distinct separate existence.

extensible markup language (XML) attribute: Name-value pair associated with an XML element.

---

[4] IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08855, USA (http://standards.ieee.org/).
[5] The standards or products referred to in this clause are trademarks of the Institute of Electrical and Electronics Engineers, Inc.
[6] *IEEE Standards Dictionary Online* subscription is available at:
http://www.ieee.org/portal/innovate/products/standard/standards_dictionary.html
[7] Adapted from VXI-1, VXIbus System Specification, Revision 3, 24 Nov. 2003.
[8] Adapted from Extensible Markup Language (XML) 1.0 (Fifth Edition). This document is available from the World Wide Web Consortium (W3C®) (http://www.w3.org/xml).

extensible markup language (XML) document: A data object that conforms to the XML requirements for being well formed. In addition, the data object is valid if it additionally conforms to semantic rules of the XML schema.

extensible markup language (XML) schema: The definition of a class of XML document, typically expressed in terms of constraints on the structure and the content of documents of that class, above and beyond the basic syntax constraints imposed by XML itself.

instance document: An extensible markup language (XML) document which conforms to a particular XML schema.

instrument: Devices (chemical, electrical, hydraulic, magnetic, mechanical, optical, pneumatic) that may be utilized to test, observe, measure, monitor, alter, generate, record, calibrate, manage, or control physical properties, movements, or other characteristics.[9]

object: An object consists of state and behavior. An object stores its states in fields (variables in some programming languages) and exposes it behavior through methods (functions in some programming languages).

peak current: The rated dc current of a particular VersaModule Eurocard extensions for instrumentation (VXI) or peripheral component interconnect extensions for instrumentation (PXI) backplane voltage.[10]

synthetic instrument: A series of hardware and software components, with standardized interfaces, to generate signals or make measurements using algorithmic/numeric processing techniques.

well formed: Conforming to all extensible markup language (XML) syntax rules.

3.2 Abbreviations and acronyms

ANSI	American National Standards Institute
ATE	automatic test equipment
ATML	automatic test markup language
ATS	automatic test system
dc	direct current
DHCP	dynamic host configuration protocol
ECL	emitter coupled logic
ECLTRIG0-1	ECL triggers 0 and 1
EIA	Electronics Industry Association
GHz	gigahertz
HW	hardware
Hz	hertz
I/O	input/output
ID	identifier

[9] Adapted from MIL-STD-1309D, Definitions of Terms for Test, Measurement and Diagnostic Equipment, Naval Electronics Systems Command (ELEX-8111), Washington, DC, 12 Feb. 1992.
[10] Adapted from VXI-1, VXIbus System Specification, Revision 3, 24 Nov. 2003.

IF	intermediate frequency
kg	kilogram
lb	pound(s)
LXI®[11]	local area networks (LAN) extensions for instrumentation
PCI	peripheral component interconnect
PCIe®	peripheral component interconnect express
PCI-SIG®	peripheral component interconnect (PCI) special interest group
PXI	peripheral component interconnect (PCI) extensions for instrumentation
PXIe®	peripheral component interconnect (PCI) extensions for instrumentation express
RFI	receiver fixture interface
RS-232	Recommended Standard 232
SCC20	Standards Coordinating Committee 20
SIWG	Synthetic Instrument Working Group
SI	system of units
SW	software
TIA	Telecommunications Industry Association
TII	test information integration
TPS	test program set
TTLTRIG0-7	transistor-transistor logic (TTL) triggers 0 through 7
URL	uniform resource locator
USB	universal serial bus
UTF-8	8-bit Unicode transformation format
UUT	unit under test
VME	VersaModule Eurocard
VXI	VME extensions for instrumentation
VXI-1	VXI system specification
W3C	World Wide Web Consortium
XML	extensible markup language

[11] Registered products (indicated by ®) listed in 3.2 are given for the convenience of users of this standard and do not constitute an endorsement by the IEEE of these products. Equivalent products may be used if they can be shown to lead to the same results.

4. InstrumentDescription schema

4.1 General

Should the reader not have a general understanding of XML schemas or XML terminology, an XML Schema Tutorial [B26] is available on the World Wide Web. This tutorial will help with the general understanding of the contents of both this clause and the ATML common schemas (Annex B of IEEE Std 1671), of which this clause makes reference.

In addition to the conventions specified in 1.5.1, *c:* identifies that the type or attribute group is contained in B.1 of IEEE Std 1671, *hc:* identifies that the type or attribute group is contained in B.2 of IEEE Std 1671, and *te:* identifies that the type or attribute group is contained in B.3 of IEEE Std 1671.

4.2 Elements

4.2.1 InstrumentDescription root (or document)

Exactly one element exists, called the *root,* or *document element,* of which no part appears in the content of any other element. This root element serves as the parent for all other elements of the InstrumentDescription schema.

The InstrumentDescription schemas root element is defined as follows:

Name	Set to
Attribute form default	Unqualified (see NOTE)
Element form default	Qualified (see NOTE)
Encoding	UTF-8
Included schema	*None*
Imported schema	urn:IEEE-1671:2010:Common urn:IEEE-1671:2010:HardwareCommon urn:IEEE-1671:2010:TestEquipment
Target namespace	urn:IEEE-1671.2:2012:InstrumentDescription
Version	2.00
XML schema namespace reference	[a]
NOTE—Qualified and unqualified are described in A.3.7 of IEEE Std 1671.[12]	

[a] The namespace reference URL is: http://www.w3.org/2001/XMLSchema.

[12] Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement this standard.

4.2.1.1 InstrumentDescription

Base type: Extension of *inst:InstrumentDescription*
Properties: content complex

The *InstrumentDescription* element shall define a class or type of instrument.

4.2.1.1.1 Attributes

The *InstrumentDescription* element inherits the attributes from *InstrumentDescription complex type* (see 4.2.2).

4.2.1.1.2 Child elements

The *InstrumentDescription* element inherits the child elements from *InstrumentDescription complex type* (see 4.2.2).

4.2.2 InstrumentDescription complex type

Base type: Extension of *hc:HardwareItemDescription*
Properties: content complex

Instruments may be represented as individual entities or as logical groups that represent an overall source, sensor, or load capability.

Figure 1 illustrates the XML types inherited and the XML types (both simple and complex) that shall be defined in this standard, which together comprise the *InstrumentDescription*.

Within Figure 1, solid lined boxes indicate that the XML element shall be required, whereas dotted lined boxes indicate that the XML element may be used. The referenced subclause identifies where the definition of the element is located within this standard or IEEE Std 1671.

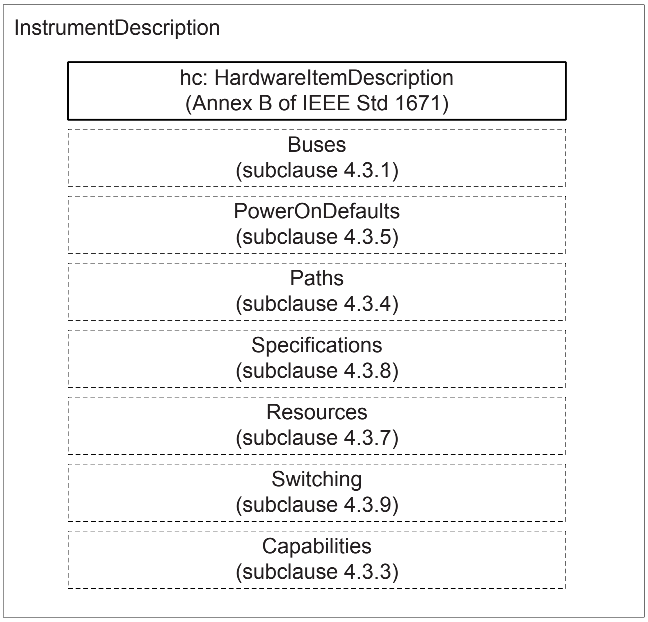

Figure 1—InstrumentDescription complex type content

4.2.2.1 Attributes

InstrumentDescription contains the following attribute, in addition to those inherited from the *hc:HardwareItemDescription* complex type and the *c:DocumentRootAttributes* attribute group defined in Annex B of IEEE Std 1671.

Name	Type	Description	Use
type	xs:string	An enumeration used to identify whether the *InstrumentDescription* is describing an “Instrument,” “Module,” or instrument “Option.”	Required

4.2.2.2 Child elements

InstrumentDescription contains the following child elements, in addition to those inherited from *hc:HardwareItemDescription* contained in Annex B of IEEE Std 1671.

Name	Type	Description	Use
Buses		See 4.3.1	Optional
Capabilities	*hc:Capabilities*	See 4.3.3	Optional
Paths	*te:Paths*	See 4.3.4	Optional
PowerOnDefaults		See 4.3.5	Optional
Resources	*hc:Resources*	See 4.3.7	Optional
Specifications	*hc:Specifications*	See 4.3.8	Optional
Switching	*hc:Switching*	See 4.3.9	Optional

4.3 Child elements

4.3.1 InstrumentDescription/Buses

Properties: isRef 0, content complex

When present, the *InstrumentDescription/Buses* child element shall document the instrument's communication bus (or buses).

4.3.1.1 Attributes

InstrumentDescription/Buses contains no attributes.

4.3.1.2 Child elements

InstrumentDescription/Buses contains the following child element:

Name	Type	Description	Use
Bus	*Bus*	See 4.3.2	Required

4.3.2 InstrumentDescription/Buses/Bus

Base type: *Bus*
Properties: isRef 0, content complex

When used, the *InstrumentDescription/Buses/Bus* child element shall document a specific bus that may control the instrument. *InstrumentDescription* supports common instrument control buses available in the marketplace, as of the publication date of this standard.

4.3.2.1 Attributes

InstrumentDescription/Buses/Bus inherits the attributes from *Bus* (see 4.4.1).

4.3.2.2 Child elements

InstrumentDescription/Buses/Bus contains no child elements.

4.3.3 InstrumentDescription/Capabilities

Base type: *hc:Capabilities*
Properties: isRef 0, content complex

When present, the *InstrumentDescription/Capabilities* child element shall document the types of signals that an instrument can produce (or measure) and the uncertainties involved in either generating or producing the signals.

This may be the ability to measure voltage, current, or resistance (such as a digital multi-meter may provide) or the ability to create and provide a clock frequency (such as a synthetic local oscillator may provide) within a synthetic instrument subsystem.

4.3.3.1 Attributes

InstrumentDescription/Capabilities contains no attributes.

4.3.3.2 Child elements

InstrumentDescription/Capabilities inherits the child elements of *hc:Capabilities* contained in Annex B of IEEE Std 1671.

4.3.4 InstrumentDescription/Paths

Base type: *te:Paths*
Properties: isRef 0, content complex

The *InstrumentDescription/Paths* child element shall identify the characteristics of the signal paths through the instrument and interface hardware.

4.3.4.1 Attributes

InstrumentDescription/Paths contains no attributes.

4.3.4.2 Child elements

InstrumentDescription/Paths inherits the child elements of *te:Paths* contained in Annex B of IEEE Std 1671.

4.3.5 InstrumentDescription/PowerOnDefaults

Properties: isRef 0, content complex

When present, the *InstrumentDescription/PowerOnDefaults* child element shall document the instrument’s power-on default state.

4.3.5.1 Attributes

InstrumentDescription/PowerOnDefaults contains no attributes.

4.3.5.2 Child elements

InstrumentDescription/PowerOnDefaults contains the following child element:

Name	Type	Description	Use
Default	*c:NamedValue*	See 4.3.6	Required

4.3.6 InstrumentDescription/PowerOnDefaults/Default

Base type: *c:NamedValue*
Properties: isRef 0, content complex

When present, the *InstrumentDescription/PowerOnDefaults/Default* child element shall document each of the parameters of the power-on default state. Each default state shall include a keyword and a value (to be interpreted as a pair).

4.3.6.1 Attributes

InstrumentDescription/PowerOnDefaults/Default inherits the attributes from *c:NamedValue* contained in Annex B of IEEE Std 1671.

4.3.6.2 Child elements

InstrumentDescription/PowerOnDefaults/Default inherits the child elements of *c:NamedValue* contained in Annex B of IEEE Std 1671.

4.3.7 InstrumentDescription/Resources

Base type: *hc:Resources*
Properties: isRef 0, content complex

When present, the *InstrumentDescription/Resources* child element shall document the physical entities within the instrument that provide source, sensor, or load capabilities.

4.3.7.1 Attributes

InstrumentDescription/Resources contains no attributes.

4.3.7.2 Child elements

InstrumentDescription/Resources inherits the child elements of *hc:Resources* contained in Annex B of IEEE Std 1671.

4.3.8 InstrumentDescription/Specifications

Base type: *hc:Specifications*
Properties: isRef 0, content complex

When present, the *InstrumentDescription/Specifications* child element shall document the specifications of an instrument.

4.3.8.1 Attributes

InstrumentDescription/Specifications contains no attributes.

4.3.8.2 Child elements

InstrumentDescription/Specifications inherits the child elements of *hc:Specifications* contained in Annex B of IEEE Std 1671.

4.3.9 InstrumentDescription/Switching

Base type: *hc:Switching*
Properties: isRef 0, content complex

When present, the *InstrumentDescription/Switching* child element shall document a signal switching instrument or a source, sensor, or load instrument that contains switching. These include: general purpose, matrix, multiplexing, coaxial, microwave, optical, and cross point switching.

4.3.9.1 Attributes

InstrumentDescription/Switching contains no attributes.

4.3.9.2 Child elements

InstrumentDescription/Switching inherits the child elements of *hc:Switching* contained in Annex B of IEEE Std 1671.

4.4 Complex types

4.4.1 Bus

Properties: abstract true
Used by child element: *InstrumentDescription/Buses/Bus*
Used by complex types: *EIA-232, Ethernet, IEEE-1394, IEEE-488, PCI, USB, VME, VXI*

When present, the *Bus* complex type shall identify the bus used to communicate with the instrument.

4.4.1.1 Attributes

Bus contains the following attribute:

Name	Type	Description	Use
defaultAddress	*c:NonBlankString*	A string that permits the instrument's default address to be indicated. Example: "255" (for VXI dynamic addressing)	Optional

4.4.1.2 Child elements

Bus contains no child elements.

4.4.2 EIA-232

Base type: Extension of *Bus*
Properties: base Bus

When present, the *EIA-232* complex type shall identify that there is an EIA-232 (formerly known as RS-232) interface to the instrument. See ANSI TIA/EIA-232-F:1997 [B1].

4.4.2.1 Attributes

EIA-232 inherits the attributes from *Bus* (see 4.4.1).

4.4.2.2 Child elements

EIA-232 contains no child elements.

4.4.3 Ethernet

Base type: Extension of *Bus*
Properties: base Bus
Used by complex type: *LXI*

When present, the *Ethernet* complex type shall identify that there is an Ethernet interface to the instrument. See IEEE Std 802.3™-2005 [B8].

4.4.3.1 Attributes

Ethernet contains the following attribute, in addition to those inherited from *Bus* (see 4.4.1).

Name	Type	Description	Use
supportsDHCP	xs:boolean	A Boolean "yes" or "no" indication as to whether the instrument's Ethernet interface supports the Dynamic Host Configuration Protocol (DHCPv4 Specification RFC 4361 [B3])	Required

4.4.3.2 Child elements

Ethernet contains no child elements.

4.4.4 IEEE-1394

Base type: Extension of *Bus*
Properties: base Bus

When present, the *IEEE-1394* complex type shall identify that there is an IEEE 1394 interface to the instrument. See IEEE Std 1394.1™-2004 [B12].

4.4.4.1 Attributes

IEEE-1394 inherits the attributes from *Bus* (see 4.4.1).

4.4.4.2 Child elements

IEEE-1394 contains no child elements.

4.4.5 IEEE-488

Base type: Extension of *Bus*
Properties: base Bus

When present, the *IEEE-488* complex type shall identify that there is an IEEE 488 interface to the instrument. See ISO 60488-2:2004 [B15] (replaces IEEE Std 488.2™).

4.4.5.1 Attributes

IEEE-488 inherits the attributes from *Bus* (see 4.4.1).

4.4.5.2 Child elements

IEEE-488 contains no child elements.

4.4.6 LXI

Base type: Extension of *Ethernet*
Properties: base Ethernet

When present, the *LXI* complex type shall identify that there is an LXI conformant Ethernet interface to the instrument. See LXI Standard [B16].

4.4.6.1 Attributes

LXI contains the following attributes, in addition to those inherited from *Ethernet* (see 4.4.3).

Name	Type	Description	Use
class	*c:NonBlankString*	The LXI class. The allowable classes are: A, B, or C.	Required
LXIVersion	*c:NonBlankString*	A string that permits the LXI version to be indicated. Example: "1.2.01"	Required

4.4.6.2 Child elements

LXI contains no child elements.

4.4.7 PCI

Base type: Extension of *Bus*
Properties: base Bus
Used by complex types: *PCIe, PXI*

When present, the *PCI* complex type shall identify that there is a PCI conformant interface to the instrument. See Conventional PCI 3.0 [B2]. Should there be more than one manufacturer, there shall be more than on instance document (one per manufacturer).

4.4.7.1 Attributes

PCI contains the following attribute, in addition to those inherited from *Bus* (see 4.4.1).

Name	Type	Description	Use
deviceID	*c:HexValue*	The identification of the particular instrument. This identifier (ID) is allocated by the vendor.	Required
vendorID	*c:HexValue*	The identification of the manufacturer of the instrument. Valid vendor IDs are allocated by the PCI Special Interest Group (PCI-SIG) to ensure uniqueness.	Required

4.4.7.2 Child elements

PCI contains no child elements.

4.4.8 PCIe

Base type: Extension of *PCI*
Properties: base PCI

When present, the *PCIe* complex type shall identify that there is a PCIe conformant interface to the instrument. See PCIe Base Specification 1.1 [B19].

4.4.8.1 Attributes

PCIe contains the following attribute, in addition to those inherited from *PCI* (see 4.4.7).

Name	Type	Description	Use
numberOfLanes	xs:int	The integer number of lanes the instrument supports. Lanes are point-to-point full duplex serial links. Each PCIe slot carries either 1, 2, 4, 8, 16, or 32 lanes of data between the motherboard and the card. Lane counts are written with an x prefix (e.g., x1 for a single-lane card and x16 for a 16-lane card).	Required

4.4.8.2 Child elements

PCIe contains no child elements.

4.4.9 PXI

Base type: Extension of *PCI*
Properties: base PCI
Used by complex type: *PXIe*

When present, the *PXI* complex type shall identify that there is a PXI conformant interface to the instrument. See PXI-1 [B20] and PXI-2 [B21].

4.4.9.1 Attributes

PXI contains the following attributes, in addition to those inherited from *PCI* (see 4.4.7).

Name	Type	Description	Use
deviceCategory	*DeviceCategory*	See 4.5.1	Required
memorySize	xs:int	The number of bits of physical memory.	Required
slots	xs:int	The number of PXI slots provided or consumed by this device. Example: “1”	Required
slotSize	xs:string	A string specifying which of the two defined sizes this PXI component supports. The string shall be one of the following: 3U or 6U.	Required
slotWeight	xs:double	For a PXI module, this is a negative floating point number representing the weight of the module in kilogram (kg) (see IEEE Std 260.1-2004™ [B7]) per slot occupied. Example: “–1.5” For a PXI mainframe, this is a positive floating point number representing the weight of the mainframe in kilogram (kg) (see IEEE Std 260.1-2004™) per slot provided. Example: “15.5” (see NOTE)	Required
NOTE—1 kg = 2.20462 lb			

4.4.9.2 Child elements

PXI contains the following child elements:

Name	Type	Description	Use
DynamicCurrent	*PXIBackplaneVoltages*	See 4.4.10	Required
PeakCurrent	*PXIBackplaneVoltages*	See 4.4.11	Required
SupportedClockSources	*SupportedClockSources*	See 4.4.12	Required

4.4.10 PXI/DynamicCurrent

Base type: *PXIBackplaneVoltages*
Properties: isRef 0, content complex

When present, the *PXI/DynamicCurrent* child element shall document the dynamic current (expressed in amperes) consumed or generated on each of the discrete PXI backplane voltages.

4.4.10.1 Attributes

PXI/DynamicCurrent inherits the attributes from *PXIBackplaneVoltages* (see 4.4.13).

4.4.10.2 Child elements

PXI/DynamicCurrent contains no child elements.

4.4.11 PXI/PeakCurrent

Base type: *PXIBackplaneVoltages*
Properties: isRef 0, content complex

When present, the *PXI/PeakCurrent* child element shall document the peak current (expressed in amperes) consumed or generated on each of the discrete PXI backplane voltages.

4.4.11.1 Attributes

PXI/PeakCurrent inherits the attributes from *PXIBackplaneVoltages* (see 4.4.13).

4.4.11.2 Child elements

PXI/PeakCurrent contains no child elements.

4.4.12 PXI/SupportedClockSources

Base type: *SupportedClockSources*
Properties: isRef 0, content complex

When present, the *PXI/SupportedClockSources* child element shall document the clock sources the instrument supports.

4.4.12.1 Attributes

PXI/SupportedClockSources inherits the attributes from *SupportedClockSources* (see 4.4.15).

4.4.12.2 Child elements

PXI/SupportedClockSources contains no child elements.

4.4.13 PXIBackplaneVoltages

Used by child elements: *PXI/DynamicCurrent, PXI/PeakCurrent*

When present, the *PXIBackplaneVoltages* complex type shall store the amount of current (expressed in amperes) consumed by or provided by the device.

4.4.13.1 Attributes

PXIBackplaneVoltages contains the following attributes:

Name	Type	Description	Use
minus_12	xs:double	The amount of current on the –12 line. A positive value means the device provides current. A negative value means the device consumes current. Example: "–0.5"	Required
plus_3.3	xs:double	The amount of current on the +3.3 line. A positive value means the device provides current. A negative value means the device consumes current. Example: "–0.5"	Required
plus_5	xs:double	The amount of current on the +5 line. A positive value means the device provides current. A negative value means the device consumes current. Example: "–0.5"	Required
plus_12	xs:double	The amount of current on the +12 line. A positive value means the device provides current. A negative value means the device consumes current. Example: "–0.5"	Required

4.4.13.2 Child elements

PXIBackplaneVoltages contains no child elements.

4.4.14 PXIe

Base type: Extension of *PXI*
Properties: base PXI

When present, the *PXIe* complex type shall identify that there is a PXI Express conformant interface to the instrument. See PXI-5 [B22] and PXI-6 [B23].

4.4.14.1 Attributes

PXIe contains the following attribute, in addition to those inherited from *PXI* (see 4.4.9).

Name	Type	Description	Use
numberOfLanes	xs:int	The integer number of lanes the instrument supports. Lanes are point-to-point full duplex serial links. Each PCIe slot carries either 1, 2, 4, 8, 16, or 32 lanes of data between the motherboard and the card. Lane counts are written with an x prefix (e.g., x1 for a single-lane card and x16 for a 16-lane card).	Required

4.4.14.2 Child elements

PXIe inherits the child elements of *PXI* (see 4.4.9).

4.4.15 SupportedClockSources

Used by complex types: *PXI/SupportedClockSources*, *VXI/SupportedClockSources*

When present, the *SupportedClockSources* complex type shall identify which clock sources the device will accept. Each type (backplane, external, and internal) shall be indicated as either a “yes” or “no,” allowing every combination from “none” to “all” to be represented. Example: An instrument that may only be clocked from an external source would have the backplane and internal attributes set to “0” (no) and the external attribute set to “1” (yes).

4.4.15.1 Attributes

SupportedClockSources contains the following attributes:

Name	Type	Description	Use
backplane	xs:boolean	Whether or not the device will accept a backplane clock. Example: “1” = “yes”	Required
external	xs:boolean	Whether or not the device will accept an external clock. Example: “0” = “no”	Required
internal	xs:boolean	Whether or not the device will accept an internal clock. Example: “1” = “yes”	Required

4.4.15.2 Child elements

SupportedClockSources contains no child elements.

4.4.16 USB

Base type: Extension of *Bus*
Properties: base Bus

When present, the *USB* complex type shall identify that there is a USB conformant interface to the instrument. See Universal Serial Bus [B24].

4.4.16.1 Attributes

USB inherits the attributes from *Bus* (see 4.4.1).

4.4.16.2 Child elements

USB contains the following child element:

Name	Type	Description	Use
Version	*hc:VersionIdentifier*	See 4.4.17	Optional

4.4.17 USB/Version

Base type: *hc:VersionIdentifier*
Properties: isRef 0, content complex

When present, the *USB/Version* child element shall identify the version of USB supported.

4.4.17.1 Attributes

USB/Version inherits the attributes from the *hc:VersionIdentifier* complex type contained in IEEE Std 1671, Annex B.

4.4.17.2 Child elements

USB/Version contains no child elements.

4.4.18 VME

Base type: Extension of *Bus*
Properties: base Bus

When present, the *VME* complex type shall identify that there is an IEEE Std 1014™ (VME) conformant interface (either 32-bit or 64-bit VME) to the instrument. See IEEE Std 1014-1987 [B9].

4.4.18.1 Attributes

VME inherits the attributes from *Bus* (see 4.4.1).

4.4.18.2 Child elements

VME contains no child elements.

4.4.19 VXI

Base type: Extension of *Bus*
Properties: base Bus

When present, the *VXI* complex type shall identify that there is an IEEE Std 1155™ conformant interface to the instrument. See IEEE Std 1155-1992 [B11].

4.4.19.1 Attributes

VXI contains the following attributes, in addition to those inherited from *Bus* (see 4.4.1).

Name	Type	Description	Use
addressSpace	xs:string	Permits the instrument's VXI address space to be identified, as an integer value that is one of the following: 0 = A16/A24 1 = A16/A32 2 = A16 only	Required
deviceCategory	inst:DeviceCategory	Permits the VXI instrument category to be identified. See 4.5.1	Required
deviceClass	xs:string	Permits the VXI instrument class to be identified, as an value that is one of the following: Message = Message based Register = Register based	Required
dynamicallyConfigured	xs:boolean	For a deviceCategory = "InstrumentModule," this shall be set to "true" (1) if the VXI instrument module is dynamically configurable, or to "false" (0) otherwise. For a deviceCategory = "Slot0Device," this shall be set to "true" (1) if the slot0 device has a resource manager that supports dynamic configuration, or to "false" (0) otherwise.	Required
interruptLines	xs:int	This is an integer number representing the number of lines produced by a mainframe (a positive number) or the number of VXI interrupt lines the instrument utilizes (a negative number). Example: "–1"	Required
manufacturerID	*c:HexValue*	A 12-bit integer value, representing the HW vendor. The value is as defined by in VXI-1 [B25]. Example: "0xEDF"	Required
modelCode	*c:HexValue*	A unique 12 (or 16) bit integer value assigned by the manufacturer to represent the VXI component. Example: "0x900"	Required
requiredMemory	*c:HexValue*	A 4-bit encoded number defined in VXI-1, C.2.1.1.2 [B25]. Example: "0x8"	Required
slots	xs:int	The number of VXI slots provided or consumed by this device. Example: "1"	Required
slotSize	xs:string	A string specifying which of the four defined sizes this VXI component supports. The string shall be one of the following: A, B, C, or D. The four sizes are as defined in VXI-1 [B25]. Example: "C"	Required
slotWeight	xs:double	For a VXI module, this is a negative floating point number representing the weight of the module in kilogram (kg) per IEEE Std 260.1™-2004 [B7] per slot occupied. Example: "–1.5" For a VXI mainframe, this is a positive floating point number representing the weight of the mainframe in kilogram (kg) per IEEE Std 260.1-2004 per slot provided. Example: "15.5" (see NOTE)	Required
NOTE—1 kg = 2.20462 lb			

4.4.19.2 Child elements

VXI contains the following child elements:

Name	Type	Description	Use
DynamicCurrent	*VXIBackplaneVoltages*	See 4.4.20	Required
ECLTrigger	*VXITriggerLines*	See 4.4.21	Required
Keying		See 4.4.22	Required
ModuleCooling		See 4.4.23	Required
PeakCurrent	*VXIBackplaneVoltages*	See 4.4.24	Required
SupportedClockSources	*SupportedClockSources*	See 4.4.25	Required
TTLTrigger	*VXITriggerLines*	See 4.4.26	Required

4.4.20 VXI/DynamicCurrent

Base type: *VXIBackplaneVoltages*
Properties: isRef 0, content complex

When present, the *VXI/DynamicCurrent* child element shall identify the dynamic electrical current requirements of the VXI instrument for each of the discrete eight VXI backplane voltages. The current shall be expressed in amperes.

4.4.20.1 Attributes

VXI/DynamicCurrent inherits the attributes from *VXIBackplaneVoltages* (see 4.4.27).

4.4.20.2 Child elements

VXI/DynamicCurrent contains no child elements.

4.4.21 VXI/ECLTrigger

Base type: *VXITriggerLines*
Properties: isRef 0, content complex

When present, the *VXI/ECLTrigger* child element shall identify the number of concurrent ECL trigger lines (ECLTRIG0-1) either being outputted by, or sensed by, the VXI instrument. The number of concurrent trigger lines shall be expressed as an integer number.

4.4.21.1 Attributes

VXI/ECLTrigger inherits the attributes from *VXITriggerLines* (see 4.4.28).

4.4.21.2 Child elements

VXI/ECLTrigger contains no child elements.

4.4.22 VXI/Keying

Properties: isRef 0, content complex

When present, the *VXI/Keying* child element shall identify the VXI local bus lockout key details utilized by the instrument. Keying class 1 through 6 shall be as defined by VXI-1, B.7.2.6 [B25]. Keying class 7 shall be defined as “no local bus,” keying class 8 shall be defined as “sensor ±16 V,” and keying class 9 shall be defined as “sensor ±42 V.”

4.4.22.1 Attributes

VXI/Keying contains the following attributes:

Name	Type	Description	Use
bottomLeft	xs:int	An integer representing the keying class (1 through 6) defined by VXI-1, Figure B.28 [B25]. Classes 7, 8, and 9 are defined as follows: Class 7 = No local bus Class 8 = Sensor ±16 V Class 9 = Sensor ±42 V Bottom key supports D size modules. Bottom key on C size modules shall be specified as 7.	Required
bottomRight	xs:int	An integer representing the keying class (1 through 6) defined by the VXI-1, Figure B.28 [B25]. Classes 7, 8, and 9 are defined as follows: Class 7 = No local bus Class 8 = Sensor ±16 V Class 9 = Sensor ±42 V Bottom key supports D size modules. Bottom key on C size modules shall be specified as 7.	Required
topLeft	xs:int	An integer representing the keying class (1 through 6) defined by VXI-1, Figure B.28 [B25]. Classes 7, 8, and 9 are defined as follows: Class 7 = No local bus Class 8 = Sensor ±16 V Class 9 = Sensor ±42 V Top key supports C size modules only.	Required
topRight	xs:int	An integer representing the keying class (1 through 6) defined by VXI-1, Figure B.28 [B25]. Classes 7, 8, and 9 are defined as follows: Class 7 = No local bus Class 8 = Sensor ±16 V Class 9 = Sensor ±42 V Top key supports C size modules only.	Required

4.4.22.2 Child elements

VXI/Keying contains no child elements.

4.4.23 VXI/ModuleCooling

Properties: isRef 0, content complex

When present, the *VXI/ModuleCooling* child element shall identify the VXI instruments air-flow and back-pressure cooling characteristics.

4.4.23.1 Attributes

VXI/ModuleCooling contains the following attributes:

Name	Type	Description	Use
airflow	xs:double	Module cooling as air flow expressed in liter per second (L/s) (see IEEE Std 260.1-2004 [B7]). This value shall always be negative (as there are only "consumers," no "producers"). Example: "–5.0"	Required
backPressure	xs:double	Module cooling as back pressure expressed in pascal (Pa) (see IEEE Std 260.1-2004 [B7]). This value shall always be negative (as there are only "consumers," no "producers"). Example: "–5.0" (see NOTE)	Required

NOTE—1 mm of water is equal to 9.655 Pa.

4.4.23.2 Child elements

VXI/ModuleCooling contains no child elements.

4.4.24 VXI/PeakCurrent

Base type: *VXIBackplaneVoltages*
Properties: isRef 0, content complex

When present, the *VXI/PeakCurrent* child element shall identify the peak electrical current requirements of the VXI instrument for each of the eight discrete VXI backplane voltages. The electrical current shall be expressed in amperes.

4.4.24.1 Attributes

VXI/PeakCurrent inherits the attributes from *VXIBackplaneVoltages* (see 4.4.27).

4.4.24.2 Child elements

VXI/PeakCurrent contains no child elements.

4.4.25 VXI/SupportedClockSources

Base type: *SupportedClockSources*
Properties: isRef 0, content complex

When present, the *VXI/SupportedClockSources* child element shall document the clock sources the instrument supports.

4.4.25.1 Attributes

VXI/SupportedClockSources inherits the attributes from *SupportedClockSources* (see 4.4.15).

4.4.25.2 Child elements

VXI/SupportedClockSources contains no child elements.

4.4.26 VXI/TTLTrigger

Base type: *VXITriggerLines*
Properties: isRef 0, content complex

When present, the *VXI/TTLTrigger* child element shall identify the number of concurrent TTL trigger lines (TTLTRIG0-7) either being outputted by, or sensed by, the VXI instrument. The number of concurrent trigger lines shall be expressed as an integer number.

4.4.26.1 Attributes

VXI/TTLTrigger inherits the attributes from *VXITriggerLines* (see 4.4.28).

4.4.26.2 Child elements

VXI/TTLTrigger contains no child elements.

4.4.27 VXIBackplaneVoltages

Used by child elements: *VXI/DynamicCurrent, VXI/PeakCurrent*

When present, the *VXIBackplaneVoltages* complex type shall store the amount of electrical current sunk by the VXI instrument on each of the eight discrete VXI backplane voltages. The electrical current shall be expressed in amperes.

4.4.27.1 Attributes

VXIBackplaneVoltages contains the following attributes:

Name	Type	Description	Use
minus_2	xs:double	The amount of current on the –2 V line. A positive value means the device provides current. A negative value means the device consumes current. Example: “–0.5”	Required
minus_5.2	xs:double	The amount of current on the –5.2 V line. A positive value means the device provides current. A negative value means the device consumes current. Example: “–0.5”	Required
minus_12	xs:double	The amount of current on the –12 V line. A positive value means the device provides current. A negative value means the device consumes current. Example: “–0.5”	Required
minus_24	xs:double	The amount of current on the –24 V line. A positive value means the device provides current. A negative value means the device consumes current. Example: “–0.5”	Required
plus_5	xs:double	The amount of current on the +5 V line. A positive value means the device provides current. A negative value means the device consumes current. Example: “–0.5”	Required
plus_5_standby	xs:double	The amount of current on the +5 V standby line. A positive value means the device provides current. A negative value means the device consumes current. Example: “–0.5”	Required
plus_12	xs:double	The amount of current on the +12 V line. A positive value means the device provides current. A negative value means the device consumes current. Example: “–0.5”	Required
plus_24	xs:double	The amount of current on the +24 V line. A positive value means the device provides current. A negative value means the device consumes current. Example: “–0.5”	Required

4.4.27.2 Child elements

VXIBackplaneVoltages contains no child elements.

4.4.28 VXITriggerLines

Used by child elements: *VXI/ECLTrigger, VXI/TTLTrigger*

When present, the *VXITriggerLines* complex type shall identify the number of concurrent trigger lines either being outputted by, or sensed by, the VXI instrument. The number of concurrent trigger lines shall be expressed as an integer number.

4.4.28.1 Attributes

VXITriggerLines contains the following attributes:

Name	Type	Description	Use
sense	xs:int	The number of VXI triggers that may concurrently receive inputs. Allowable values are 0 or greater.	Optional
source	xs:int	The number of VXI triggers that may concurrently generate outputs. Allowable values are 0 or greater.	Optional

4.4.28.2 Child elements

VXITriggerLines contains no child elements.

4.5 Simple types

4.5.1 DeviceCategory

Type: restriction of xs:string
Used by child elements: *VXI/@deviceCategory, PXI/@deviceCategory*

Permits the instrument category to be identified, as one of the following: "InstrumentModule," "Slot0Device," or "Mainframe."

5. InstrumentDescription instance schema

Should the reader not have a general understanding of XML schemas or XML terminology, an *XML Schema Tutorial* [B26] is available on the World Wide Web. This tutorial will help with the understanding of the contents of both this clause and the ATML common schemas (Annex B of IEEE Std 1671), of which this clause makes reference.

In addition to the conventions specified in 1.5.1, "*c:*" identifies that the type or attribute group is contained in B.1 of IEEE Std 1671, "*hc:*" identifies that the type or attribute group is contained in B.2 of IEEE Std 1671, and "*te:*" identifies that the type or attribute group is contained in B.3 of IEEE Std 1671.

5.1 Elements

5.1.1 InstrumentInstance root (or document)

Exactly one element exists, called the *root* or *document element,* of which no part appears in the content of any other element. This root element serves as the parent for all other elements of the *InstrumentInstance* schema.

The *InstrumentInstance* schemas root element is defined as follows:

Name	Set to
Attribute form default	Unqualified (see NOTE)
Element form default	Qualified (see NOTE)
Encoding	UTF-8
Included schema	None
Imported schema	urn:IEEE-1671:2010:Common urn:IEEE-1671:2010:HardwareCommon urn:IEEE-1671:2010:TestEquipment
Target namespace	urn:IEEE-1671.2:2012:InstrumentInstance
Version	2.00
XML schema namespace reference	[a]
NOTE—Qualified and unqualified are described in A.3.7 of IEEE Std 1671.	

[a] The namespace reference URL is: http://www.w3.org/2001/XMLSchema.

5.1.2 InstrumentInstance

Base type: Extension of *instr:InstrumentInstance*
Properties: content complex

The *InstrumentInstance* element shall identify a specific instrument.

5.1.2.1 Attributes

InstrumentInstance inherits the attributes from *InstrumentInstance complex type* (See 5.2.1).

5.1.2.2 Child elements

InstrumentInstance inherits the child elements from *InstrumentInstance complex type* (See 5.2.1).

5.2 Complex types

5.2.1 InstrumentInstance

Base type: Extension of *c:HardwareInstance*
Properties: content complex

The *InstrumentInstance* element shall identify a specific instrument.

5.2.1.1 Attributes

InstrumentInstance contains the following attributes in addition to both the s*ecurityClassification* and *uuid* attributes from the *DocumentRootAttributes* attribute group defined in Annex B of IEEE Std 1671.

Name	Type	Description	Use
name	*c:NonBlankString*	A string that permits the name of the instance document to be indicated.	Optional
version	*c:NonBlankString*	A string that permits the version of the instance document to be indicated.	Optional

5.2.1.2 Child elements

InstrumentInstance contains the following child element, in addition to those inherited from the *c:HardwareInstance* complex type contained in Annex B of IEEE Std 1671.

Name	Type	Description	Use
Paths	*te:Paths*	See 5.2.4	Optional
Capabilities	*hc:Capabilities*	See 5.2.2	Optional
Extension	*c:Extension*	See 5.2.3	Optional

5.2.2 InstrumentInstance/Capabilities

Base type: *hc:Capabilities*
Properties: isRef 0 content complex

The *InstrumentInstance/Capabilities* child element shall identify the capabilities of a specific instrument.

5.2.2.1 Attributes

InstrumentInstance/Capabilities contains no attributes.

5.2.2.2 Child elements

InstrumentInstance/Capabilities inherits the child elements of the *hc:Capabilities* complex type contained in Annex B of IEEE Std 1671.

5.2.3 InstrumentInstance/Extension

Base type: *c:Extension*
Properties: isRef 0, content complex

The *InstrumentInstance/Extension* child element shall provide a specific extension point for use cases that require elements not provided in the basic structure.

5.2.4 InstrumentInstance/Paths

Base type: *te:Paths*
Properties: isRef 0, content complex

The *InstrumentInstance/Paths* child element shall identify the characteristics of the signal paths through the specific instrument and interface hardware.

5.2.4.1 Attributes

InstrumentInstance/Paths contains no attributes.

5.2.4.2 Child elements

InstrumentInstance/Paths inherits the child elements of *te:Paths* contained in Annex B of IEEE Std 1671.

5.3 Simple types

None.

6. ATML InstrumentDescription XML schema names and locations

The IEEE provides a download Web site for material published in association with published IEEE Standards, presented in machine-friendly format. This material is digital rights management restricted use material. The ATML family of standards utilizes this download Web site to allow easy accessibility to all of the ATML family XML schemas (and in some cases, example XML instance documents). As depicted by Figure 2, the IEEE download Web site (http://standards.ieee.org/downloads/) contains several folders, each folder labeled by an associated IEEE standards number (e.g., IEEE 1671 series standards are in the 1671 folder). Each folder under this "base" IEEE standard number contains the material (XML schemas, etc.) for that ATML family component standard. ATML family component standards are identified by their IEEE 1671 series "dot" standard number and the year in which that standard was published by the IEEE.

NOTE 1—Standards that are revised will contain a folder for the year in which the standard is "reissued." Both folders (for each year the standard was published) will be present on the IEEE download Web site.

NOTE 2—Folders for a particular standard are not available until the standard is published by the IEEE, and providing the standard has associated material that is to be made available via the download Web site.

Figure 2 depicts a portion of the entire IEEE download Web site, as it pertains to the InstrumentDescription ATML family standard.

The IEEE SCC20 TII subcommittee Web site (http://grouper.ieee.org/groups/scc20/tii) provides access to material not yet published as an IEEE Standard. This material is also digital rights management restricted use material. The ATML family of standards utilize this site to allow easy accessibility to any of the ATML family XML schemas (and in some cases, example XML instance documents) not yet approved by the IEEE Standards Board.

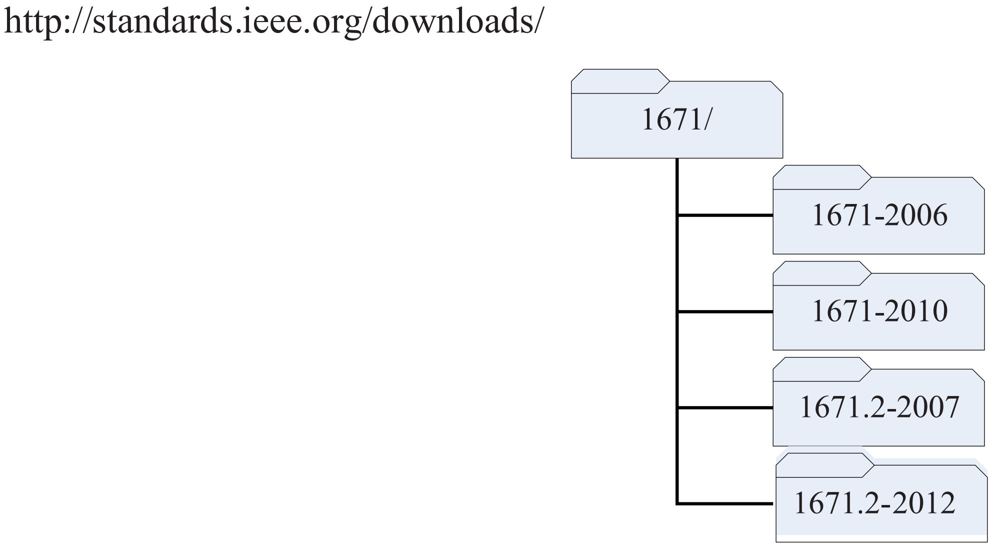

Figure 2—ATML InstrumentDescription related IEEE download Web site structure

The InstrumentDescription ATML family component standard, where the component is defined, their associated XML schemas names, and the IEEE download Web site folder name (where the XML schemas shall be located), is as defined in Table 1.

Table 1—ATML family XML schema name and folder location

Component	Defined in Clause	XML schema name	IEEE Download Website folder (see Figure 2)
InstrumentDescription	4	InstrumentationDescription.xsd	1671.2-2012
InstrumentInstance	5	InstrumentationInstance.xsd	1671.2-2012

The XML schema identified in Table 2 includes ATML common elements. The ATML common element (e.g., component), where the component is defined, the associated XML schemas name, and the IEEE download Web site folder name (where the XML schema shall be located).

Table 2—ATML Common element XML schema name and location

Component	Location defined in IEEE Std 1671-2010	XML schema name	IEEE download Website folder (see Figure 2)
Common	Annex B.1	Common.xsd	1671-2010
Hardware common	Annex B.2	HardwareCommon.xsd	1671-2010
Test equipment	Annex B.3	TestEquipment.xsd	1671-2010
Capabilities	Annex C.1	Capabilities.xsd	1671-2010
Wire lists	Annex C.2	WireLists.xsd	1671-2010

7. ATML XML schema extensibility

The provision of an extension mechanism is necessary to ensure the viability of the specification and allow producers and consumers of ATML instance documents to interoperate in those cases where there is a requirement to exchange relevant data that are not included in the *InstrumentDescription* associated XML schema. The use of the extensions shall be done in a way that ensures that a conformant consumer can utilize the extended file without error, discard, or otherwise sidestep the extended data, and use the non-extended portions of the data as they are intended, without error or loss of functionality.

Extensions shall be additional information added to the content model of the element being extended.

Extensions shall not repackage existing information entities that are already supported by the *InstrumentDescription* XML schema.

An extended instance document shall be accompanied by the extension XML schema and documentation sufficient to explain the need for the extension as well as by the underlying semantics and relationship(s) to the base *InstrumentDescription* schema.

The ATML family of standards associated XML schemas allow for three forms of extension:

a) Wildcard-based extensions allow for the extension of XML schemas with additional elements.

b) Type derivation allows for extending the set of data types by deriving a new type from an existing common element type.

c) Lists derived from *c:NamedValues* allowing user-defined properties with attached values.

8. Conformance

The minimal expectations for *InstrumentDescription* conformant XML instance documents shall be that a completely populated instance document is considered valid if said document complies with:

a) The constraints expressed in the *InstrumentDescription* schema associated with this standard and

b) Any constraints imposed by inherited elements from Annex B of IEEE Std 1671.

The minimal expectations for *InstrumentInstance* conformant XML instance documents shall be that a completely populated instance document is considered valid if said document complies with:

a) The constraints expressed in the *InstrumentInstance* schema associated with this standard and

b) Any constraints imposed by inherited elements from Annex B of IEEE Std 1671.

Extensions are permitted to both the *InstrumentDescription* and *InstrumentInstance* schemas, but shall only occur through the facility of the extensibility mechanism described in Clause 7. In short, extensions may only appear in the specific *<extension>* tags provided in the associated XML schema. As defined in the W3C XML schema standard, an extended schema must conform to the W3C XML schema specification and shall not contain any entities defined in the base schema.

Annex A

(informative)

IEEE download Web site material associated with this document

This document includes supporting material required to maintain and/or develop the ATML framework as well as maintain the ATML family of standards. This material is published by the IEEE in association with this document, presented in a machine-friendly format. This is digital rights management restricted use material. The ATML family of standards utilize this download Web site to allow easy accessibility to these documents XML schemas, and associated material referenced within this document (e.g., examples or committee drafts). For an explanation, and the location, of the IEEE download Web site and its structure (as it pertains to the ATML family of standards), see Clause 6. The material available on the IEEE download Web site in association with this document is descried in Table A.1.[13]

Table A.1—IEEE download Web site contents

File	Description
mentDescription.xsd	The ATML InstrumentDescription schema defined in Clause 4.
mentInstance.xsd	The ATML InstrumentInstance schema defined in Clause 5.
ıon.xsd	The ATML common element schema defined in IEEE Std 1671-2010 Annex B.1.
vareCommon.xsd	The ATML common schema defined in IEEE Std 1671-2010 Annex B.2.
quipment.xsd	The ATML common schema defined in IEEE Std 1671-2010 Annex B.3.
ilities.xsd	The ATML common schema defined in IEEE Std 1671-2010 Annex C.1.
ists.xsd	The ATML common schema defined in IEEE Std 1671-2010 Annex C.2.
2_Example_Capabilities_Definition.xml	Examples
2_Example_Capabilities_InstrumentDescription.xml	
2_Example_Capabilities_InstrumentInstance.xml	
2_Example_GraphAsSpecification.xml	
2_Example_InstrumentWithCalibration.xml	
2_Example_InstrumentWithCapabilities.xml	
2_Example_InstrumentWithTriggering.xml	
2_Example_MultiChannelInstrument.xml	
2_Example_Options_AddWidebandIfOutput.xml	
2_Example_Options_ConnectorChange.xml	
2_Example_Options_DeleteFrontInput.xml	
2_Example_Options_InstrumentDescription.xml	
2_Example_Options_InstrumentInstance.xml	
2_Example_Options_WidebandPreamp.xml	
2_Example_ParallelMeasurements(Traces).xml	
2_Example_PinsPortsConnectors_PowerSource.xml	
2_Example_PinsPortsConnectors_Simple.xml	
2_Example_SimpleInstrument.xml	
2_Example_SwitchInstrument.xml	
zer.xml	
Converter.xml	
eticWaveformGenerator.xml	
nverter.xml	
ne.txt	This file contains user information pertaining to the files posted, related files, and their usage.

[13] Available: http://standards.ieee.org/downloads/1671/.

Annex B

(informative)

Users information and examples

B.1 Usage within an automatic test station

The purpose of this example is to show *InstrumentDescriptions* and *InstrumentInstances*, describing the capabilities of the various elements incorporated in a simple test station.

This test station could then be specified by an ATML test station (governed by the ATML test station standard IEEE Std 1671.6™ instance document, which may utilize these instrument instance documents (.xml files) in its specification of the test station capabilities. See IEEE Std 1671.6 [B14].

Figure B.1 shows a very simple test scenario for a unit under test (UUT), where the instruments within the test station are a simple instrument, a multi-channel input/output (I/O) instrument, and switching used to route the simple instruments connections within the tester to multiple ports on a receiver fixture interface (RFI).

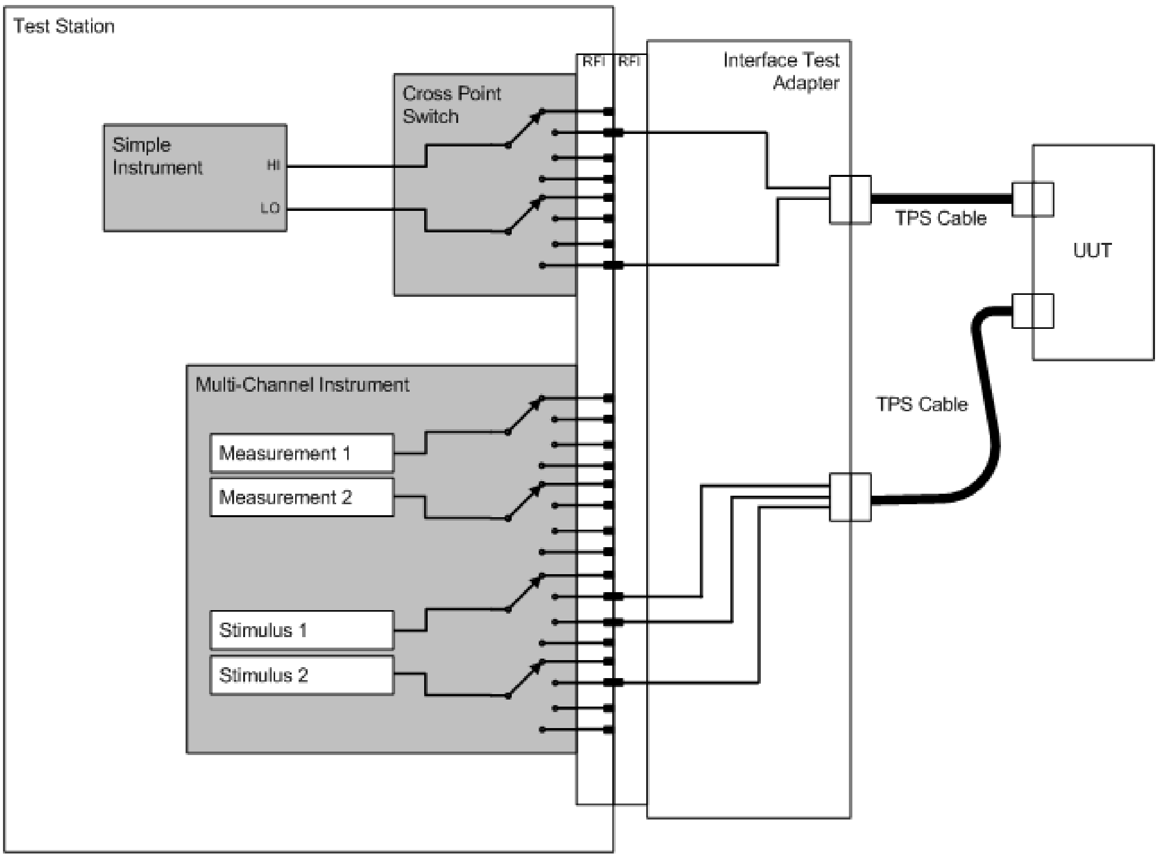

Figure B.1—Example UUT test scenario

As an example, the instance document for the multi-channel I/O instrument may be developed by either the instrument vendor or by the test station system integrator. Should the instrument vendor make an instance document available, the definition of the multi-channel I/O instrument would most likely be "from the angle of the instrument vendor" and reflect every capability advertised by the vendor. Should the test station integrator develop the instance document, it would most likely

a) reflect how the multi-channel I/O instrument is allowed to be used, and
b) reflect an "interpretation" of the multi-channel I/O instrument vendor specifications.

For the purposes of this example, the multi-channel I/O instrument instance that has been included could have been created by either the vendor or station integrator.

In support of Figure B.1, a TestStation instance document will refer to the instrument instance documents through the use of the unique IDs stored in the appropriate TestStation elements' *uuid* attributes. Within this example, the TestStation instance document would point to the following documents:

a) *1671_2_Example_SimpleInstrument.xml*—describes the simple instrument in the test station.
b) *1671_2_Example_MultiChannelInstrument.xml*—describes the multi-channel instrument in the test station.
c) *1671_2_Example_SwitchInstrument.xml*—describes the switching in the test station.

B.2 Instruments with calibration, capabilities, or triggering

These three examples provide sample instrument descriptions that include either: calibration requirements (*1671_2_Example_InstrumentWithCalibration.xml*), capabilities with reference to IEEE Std 1641™-2010 [B13] signal descriptions (*1671_2_Example_InstrumentWithCapabilities.xml*), or triggering definitions (*1671_2_Example_InstrumentWithTriggering.xml*).[14]

B.3 Instrument options

These three examples provide sample instrument descriptions that include either: add a wide-band IF output (*1671_2_Example_Options_AddWidebandIfOutput.xml*), change a connector (*1671_2_Example_Options_ConnectorChange.xml*), or to delete a front panel input (*1671_2_Example_Options_DeleteFrontInput.xml*).[15]

B.4 Capabilities

These three examples provide sample instrument description and instrument instances which include references to IEEE Std 1671 capabilities (*1671_2_Example_Capabilities_Definition.xml*), (*1671_2_Example_Capabilities_InstrumentDescription.xml*), and (*1671_2_Example_Capabilities_InstrumentInstance.xml*).[16]

---

[14] These examples are available at: http://standards.ieee.org/downloads/1671/1671.2-2012/.
[15] See Footnote 14.
[16] See Footnote 14.

B.5 Pins, ports, and connectors

These two examples provide sample instrument descriptions that include either: power source connectors (*1671_2_Example_PinsPortsConnectors_PowerSource.xml*); or a simple pin, port, and connector (*1671_2_Example_PinsPortsConnectors.xml*).[17]

B.6 Specifications represented by graphs

This example provides a sample instrument description that includes a specification represented by a graph (*1671_2_Example_GraphAsSpecification.xml*).[18]

B.7 Parallel measurements (traces)

This example provides a sample instrument description that includes the description of parallel measurements (*1671_2_Example_ParallelMeasurements(Traces).xml*).[19]

[17] See Footnote 14.
[18] See Footnote 14.
[19] See Footnote 14.

Annex C

(informative)

Glossary

For the purposes of this standard, the following terms and definitions apply. These and other terms within IEEE standards are found in The *IEEE Standards Dictionary Online* [B5][20].

attribute: A documenting characteristic of an entity.

automatic test equipment (ATE): A system providing a test capability for the automatic testing of one or more units under test (UUTs). The ATE system consists of a controller, test resource devices, and peripherals. The controller directs the testing process and interprets the results. The test resource devices provide stimuli, measurements, and physical interconnections. The peripherals, such as displays, keyboards, printers, mass storage, etc., supply the necessary capability for information management.

automatic test system (ATS): Includes the automatic test equipment (ATE) as well as all support equipment, software, test program (TP), and adapters.

framework: A collection of classes created specifically to serve the needs of an application area.

test program set (TPS): The complete set of hardware, software, and documentation needed to evaluate a unit under test UUT on a given test system.

unit under test (UUT): The entity to be tested. It may range from a simple component to a complete system.

[20] See Footnote 6

Annex D

(informative)

Bibliography

Bibliographical references are resources that provide additional or helpful material but do not need to be understood or used to implement this standard. Reference to these resources is made for informational use only.

[B1] ANSI TIA/EIA-232-F:1997 (Reaff 2002), Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange.[21]

[B2] Conventional PCI 3.0.[22]

[B3] DHCPv4 Specification RFC 4361, Node Specific Client Identifiers for Dynamic Host Configuration Protocol, Version 4, Feb. 2006.[23]

[B4] Extensible Markup Language (XML) 1.0 (Fifth Edition), World Wide Web Consortium Recommendation 26, Nov. 2008.[24]

[B5] *IEEE Standards Dictionary Online*.[25]

[B6] *IEEE Standards Style Manual.*[26]

[B7] IEEE Std 260.1™-2004, IEEE Standard Letter Symbols for Units of Measurement (SI Units, Customary Inch-Pound Units, and Certain Other Units).[27,28]

[B8] IEEE Std 802.3™-2005, IEEE Standard for Information Technology—Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Networks—Specific Requirements: Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications.

[B9] IEEE Std 1014™-1987 (Reaff 2008), IEEE Standard for a Versatile Backplane Bus: VMEbus.

[B10] IEEE Std 1057™-2007, IEEE Standard for Digitizing Waveform Recorders.

[B11] IEEE Std 1155™-1992, IEEE Standard for VMEbus Extensions for Instrumentation: VXIbus.

[B12] IEEE Std 1394.1™-2004, IEEE Standard for High Performance Serial Bus Bridges.

[B13] IEEE Std 1641™-2010, IEEE Standard for Signal and Test Definition.

---

[21] ANSI publications are available from the Sales Department, American National Standards Institute, 25 West 43rd Street, 4th floor, New York, NY 10036, USA (http://www.ansi.org/).
[22] Available from the World Wide Web: http://www.pcisig.com/specifications/conventional/pci_30/.
[23] Available from the World Wide Web: http://www.isc.org/software/dhcp.
[24] Available from the World Wide Web: http://www.w3.org/TR/REC-xml/.
[25] See Footnote 6.
[26] Available from the World Wide Web: http://standards.ieee.org/guides/style/index.html.
[27] IEEE publications are available from the Institute of Electrical and Electronics Engineers Inc., 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards.ieee.org/).
[28] The standards or products referred to in this clause are trademarks of the Institute of Electrical and Electronics Engineers, Inc.

[B14] IEEE Std 1671.6™, IEEE Standard for Automatic Test Markup Language (ATML) for Exchanging Automatic Test Equipment and Test Information via XML, Exchanging Test Station Information.

[B15] ISO 60488-2:2004, Standard digital interface for programmable instrumentation—Part 2: Codes, Formats, Protocols and Common Commands [replaced IEEE Std 488.2™-1992].[29]

[B16] LXI Standard, LXI Consortium.[30]

[B17] MIL-PRF-55310D, General Specification For Oscillator, Crystal Controlled, U.S. Army CECOM, Fort Monmouth, NJ, 25 Mar. 1998.[31]

[B18] MIL-STD-1309D, Definitions of Terms for Test, Measurement and Diagnostic Equipment, Naval Electronics Systems Command (ELEX-8111), Washington, DC, 12 Feb. 1992.[32]

[B19] PCIe Base Specification 1.1.[33]

[B20] PXI-1, PXI Hardware Specification, Rev 2.2.[34]

[B21] PXI-2, PXI Software Specification, Rev 2.1.[35]

[B22] PXI-5, PXI Express Hardware Specification.[36]

[B23] PXI-6, PXI Express Software Specification.[37]

[B24] Universal Serial Bus, Revision 2.0 Specification.[38]

[B25] VXI-1, VXIbus System Specification, Revision 3, 24 Nov. 2003.[39]

[B26] XML Schema Tutorial.[40]

[29] ISO publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse (http://www.iso.ch/). ISO publications are also available in the United States from the Sales Department, American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (http://www.ansi.org/).
[30] Available from the World Wide Web: http://www.lxistandard.org/.
[31] Available from the Word Wide Web: http://standards.gsfc.nasa.gov/reviews/mil-prf-55310d/mil-prf-55310.pdf. MIL publications are available from Customer Service, Defense Printing Service, 700 Robbins Ave., Bldg. 4D, Philadelphia, PA 19111-5094.
[32] Available from the Word Wide Web: http://www.everyspec.com/MIL-STD/MIL-STD-1300-1399/download.php?spec=MIL-STD-1309D.033955.pdf.
[33] Available from the World Wide Web: http://www.pcisig.com/specifications/ordering_information/.
[34] Available from the World Wide Web: http://www.pxisa.org/Specifications/Default.aspx.
[35] Available from the World Wide Web: http://www.pxisa.org/Specifications.html.
[36] Available from the World Wide Web: http://www.pxisa.org/Specifications.html.
[37] Available from the World Wide Web: http://www.pxisa.org/Specifications.html.
[38] Available from the World Wide Web: http://www.usb.org/developers/docs/.
[39] Available from the World Wide Web: http://www.vxibus.org/.
[40] Available from the World Wide Web: http://www.xfront.com/xml-schema.html and http://www.xfront.com/xml-schema-1-1/index.html.

Annex E

(informative)

IEEE List of Participants

At the time this IEEE standard was completed, the Test Information Integration Working Group had the following membership:

Chris Gorringe, *Co-chair*
Teresa Lopes, *Co-chair*

Malcolm Brown
Matt Cornish
James Dumser
Robert Fox
Brit Frank
Chris Gorringe
Anand Jain
Dexter Kennedy
Teresa Lopes
Scott Misha
Dan Pleasant
Hugh Pritchett
Mike Seavey
Rob Spinner
John Stabler
Ronald Taylor
Josh Widzer

The following members of the individual balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Martin Baur
Bill Brown
Malcolm Brown
Keith Chow
James Dumser
Sourav Dutta
William Frank
Chris Gorringe
Randall Groves
Werner Hoelzl
Blackstock Hulme
AshleyMorris
Noriyuki Ikeuchi
Anand Jain
Teresa Lopes
Greg Luri
William Maciejewski
Mukund Modi
Ion Neag
Leslie Orlidge
K. James Phillips
Ulrich Pohl
Peter Richardson
Bartien Sayogo
Mike Seavey
Gil Shultz
Joseph Stanco
Walter Struppler
Ronald Taylor
John Vergis
Oren Yuen
Daidi Zhong

When the IEEE-SA Standards Board approved this standard on 5 December 2012, it had the following membership:

Richard H. Hulett, *Chair*
John Kulick, *Vice Chair*
Robert M. Grow, *Past Chair*
Konstantinos Karachalios, *Secretary*

Satish Aggarwal
Masayuki Ariyoshi
Peter Balma
William Bartley
Ted Burse
Clint Chaplin
Wael William Diab
Jean-Philippe Faure
Alexander Gelman
Paul Houzé
Jim Hughes
Joseph L. Koepfinger*
David J. Law
Thomas Lee
Hung Ling
Oleg Logvinov
Ted Olsen
Gary Robinson
Jon Walter Rosdahl
Sam Sciacca
Mike Seavey
Yatin Trivedi
Phil Winston
Don Wright

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Richard DeBlasio, *DOE Representative*
Michael Janezic, *NIST Representative*

Patrick Gibbons
IEEE Standards Program Manager, Document Development

Kathryn Bennett
IEEE Standards Program Manager, Technical Program Development

This page deliberately left blank

British Standards Institution (BSI)

BSI is the national body responsible for preparing British Standards and other standards-related publications, information and services.

BSI is incorporated by Royal Charter. British Standards and other standardization products are published by BSI Standards Limited.

About us

We bring together business, industry, government, consumers, innovators and others to shape their combined experience and expertise into standards -based solutions.

The knowledge embodied in our standards has been carefully assembled in a dependable format and refined through our open consultation process. Organizations of all sizes and across all sectors choose standards to help them achieve their goals.

Information on standards

We can provide you with the knowledge that your organization needs to succeed. Find out more about British Standards by visiting our website at bsigroup.com/standards or contacting our Customer Services team or Knowledge Centre.

Buying standards

You can buy and download PDF versions of BSI publications, including British and adopted European and international standards, through our website at bsigroup.com/shop, where hard copies can also be purchased.

If you need international and foreign standards from other Standards Development Organizations, hard copies can be ordered from our Customer Services team.

Subscriptions

Our range of subscription services are designed to make using standards easier for you. For further information on our subscription products go to bsigroup.com/subscriptions.

With **British Standards Online (BSOL)** you'll have instant access to over 55,000 British and adopted European and international standards from your desktop. It's available 24/7 and is refreshed daily so you'll always be up to date.

You can keep in touch with standards developments and receive substantial discounts on the purchase price of standards, both in single copy and subscription format, by becoming a **BSI Subscribing Member**.

PLUS is an updating service exclusive to BSI Subscribing Members. You will automatically receive the latest hard copy of your standards when they're revised or replaced.

To find out more about becoming a BSI Subscribing Member and the benefits of membership, please visit bsigroup.com/shop.

With a **Multi-User Network Licence (MUNL)** you are able to host standards publications on your intranet. Licences can cover as few or as many users as you wish. With updates supplied as soon as they're available, you can be sure your documentation is current. For further information, email bsmusales@bsigroup.com.

Revisions

Our British Standards and other publications are updated by amendment or revision.

We continually improve the quality of our products and services to benefit your business. If you find an inaccuracy or ambiguity within a British Standard or other BSI publication please inform the Knowledge Centre.

Copyright

Useful Contacts:

Customer Services
Tel: +44 845 086 9001
Email (orders): orders@bsigroup.com
Email (enquiries): cservices@bsigroup.com

Subscriptions
Tel: +44 845 086 9001
Email: subscriptions@bsigroup.com

Knowledge Centre
Tel: +44 20 8996 7004
Email: knowledgecentre@bsigroup.com

Copyright & Licensing
Tel: +44 20 8996 7070
Email: copyright@bsigroup.com

BSI Group Headquarters

389 Chiswick High Road London W4 4AL UK

bsi.