

ISO/TC 205

Secretariat: **ANSI**

Voting begins on:  
**2008-11-20**

Voting terminates on:  
**2009-01-20**

---

---

## Building automation and control systems (BACS) —

### Part 6: Data communication conformance testing

*Systèmes d'automatisation et de gestion technique du bâtiment —  
Partie 6: Essais de conformité de la communication de données*

**iTeh STANDARD PREVIEW**  
(standards.iteh.ai)  
Full standard:  
<https://standards.iteh.ai/catalog/standards/sis/5b942d0f-f0f4-48ca-aa86-dec20e2017/iso-16484-6-2008>

Please see the administrative notes on page iii

RECIPIENTS OF THIS DRAFT ARE INVITED TO SUBMIT, WITH THEIR COMMENTS, NOTIFICATION OF ANY RELEVANT PATENT RIGHTS OF WHICH THEY ARE AWARE AND TO PROVIDE SUPPORTING DOCUMENTATION.

IN ADDITION TO THEIR EVALUATION AS BEING ACCEPTABLE FOR INDUSTRIAL, TECHNOLOGICAL, COMMERCIAL AND USER PURPOSES, DRAFT INTERNATIONAL STANDARDS MAY ON OCCASION HAVE TO BE CONSIDERED IN THE LIGHT OF THEIR POTENTIAL TO BECOME STANDARDS TO WHICH REFERENCE MAY BE MADE IN NATIONAL REGULATIONS.



Reference number  
ISO/FDIS 16484-6:2008(E)

**PDF disclaimer**

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

**iTeh STANDARD PREVIEW**  
(standards.iteh.ai)  
Full standard:  
<https://standards.iteh.ai/catalog/standards/sist/5b942d7b-f0f4-48ca-aa86-dec20e20177/iso-16484-6-2009>

**Copyright notice**

This ISO document is a Draft International Standard and is copyright-protected by ISO. Except as permitted under the applicable laws of the user's country, neither this ISO draft nor any extract from it may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, photocopying, recording or otherwise, without prior written permission being secured.

Requests for permission to reproduce should be addressed to either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office  
Case postale 56 • CH-1211 Geneva 20  
Tel. + 41 22 749 01 11  
Fax + 41 22 749 09 47  
E-mail [copyright@iso.org](mailto:copyright@iso.org)  
Web [www.iso.org](http://www.iso.org)

Reproduction may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

## ISO/CEN PARALLEL PROCESSING

The CEN Secretary-General has advised the ISO Secretary-General that this final draft International Standard covers a subject of interest to European standardization. In accordance with the ISO-lead mode of collaboration as defined in the Vienna Agreement, this final draft is hereby submitted to a parallel two-month FDIS vote in ISO and a three-month UAP vote in CEN.

**Positive votes shall not be accompanied by comments.**

**Negative votes shall be accompanied by the relevant technical reasons.**

In accordance with the provisions of Council Resolution 15/1993, this document is **circulated in the English language only**.

**iTeh STANDARD PREVIEW**  
(standards.iteh.ai)

Full standard:  
<https://standards.iteh.ai/catalog/standards/sist/5b942d7b-f0f4-48ca-aa86-dec20e20177/iso-16484-6-2009>

## Contents

CLAUSE	PAGE
1 Scope.....	1
2 Relationship between this part of ISO 16484 and ANSI/ASHRE 135.1-2007 .....	1
3 Terms, definitions and abbreviated terms .....	1
4 ELECTRONIC PICS FILE FORMAT .....	3
4.1 Character Encoding .....	3
4.2 Structure of EPICS Files .....	4
4.3 Character Strings .....	4
4.4 Notational Rules for Parameter Values .....	4
4.5 Sections of the EPICS File .....	6
5 EPICS CONSISTENCY TESTS.....	22
6 CONVENTIONS FOR SPECIFYING BACnet CONFORMANCE TESTS .....	23
6.1 TCSL Components .....	24
6.2 TCSL Statements .....	25
6.3 Time Dependencies .....	29
6.4 BACnet References .....	29
7 OBJECT SUPPORT TESTS.....	30
7.1 Read Support for Properties in the Test Database .....	30
7.2 Write Support for Properties in the Test Database .....	30
7.3 Object Functionality Tests .....	31
8 APPLICATION SERVICE INITIATION TESTS.....	99
8.1 AcknowledgeAlarm Service Initiation Tests.....	100
8.2 ConfirmedCOVNotification Service Initiation Tests .....	100
8.3 UnconfirmedCOVNotification Service Initiation Tests .....	109
8.4 ConfirmedEventNotification Service Initiation Tests .....	111
8.5 UnconfirmedEventNotification Service Initiation Tests .....	141
8.6 GetAlarmSummary Service Initiation Tests .....	147
8.7 GetEnrollmentSummary Service Initiation Tests.....	147
8.8 GetEventInformation Service Initiation Tests.....	149
8.9 LifeSafetyOperation Service Initiation Tests .....	149
8.10 SubscribeCOV Service Initiation Tests.....	150
8.11 SubscribeCOVProperty Service Initiation Tests .....	151
8.12 AtomicReadFile Service Initiation Tests.....	152
8.13 AtomicWriteFile Service Initiation Tests.....	152
8.14 AddListElement Service Initiation Tests.....	153
8.15 RemoveListElement Service Initiation Tests .....	153
8.16 CreateObject Service Initiation Tests.....	154
8.17 DeleteObject Service Initiation Tests.....	155
8.18 ReadProperty Service Initiation Tests.....	155
8.19 ReadPropertyConditional Service Initiation Tests .....	156
8.20 ReadPropertyMultiple Service Initiation Tests .....	156
8.21 ReadRange Service Initiation Tests .....	157
8.22 WriteProperty Service Initiation Tests .....	159
8.23 WritePropertyMultiple Service Initiation Tests .....	159
8.24 DeviceCommunicationControl Service Initiation Tests.....	161
8.25 ConfirmedPrivateTransfer Service Initiation Test .....	162
8.26 UnconfirmedPrivateTransfer Service Initiation Test .....	163
8.27 ReinitializeDevice Service Initiation Tests .....	163
8.28 ConfirmedTextMessage Service Initiation Tests .....	164
8.29 UnconfirmedTextMessage Service Initiation Tests .....	165
8.30 TimeSynchronization Service Initiation Tests .....	166
8.31 UTCTimeSynchronization Service Initiation Tests .....	166
8.32 Who-Has Service Initiation Tests.....	166
8.33 I-Have Service Initiation Tests.....	167

8.34	Who-Is Service Initiation Tests.....	167
8.35	I-Am Service Initiation Tests.....	168
8.36	VT-Open Service Initiation Tests.....	168
8.37	VT-Close Service Initiation Tests.....	169
8.38	VT-Data Service Initiation Tests.....	170
8.39	RequestKey Service Initiation Tests.....	172
8.40	Authenticate Service Initiation Tests.....	173
9	APPLICATION SERVICE EXECUTION TESTS.....	176
9.1	AcknowledgeAlarm Service Execution Tests.....	177
9.2	ConfirmedCOVNotification Service Execution Tests.....	189
9.3	UnconfirmedCOVNotification Service Execution Tests.....	192
9.4	ConfirmedEventNotification Service Execution Tests.....	192
9.5	UnconfirmedEventNotification Service Execution Tests.....	194
9.6	GetAlarmSummary Service Execution Tests.....	194
9.7	GetEnrollmentSummary Service Execution Tests.....	195
9.8	GetEventInformation Service Execution Tests.....	199
9.9	LifeSafetyOperation Service Execution Test.....	201
9.10	SubscribeCOV Service Execution Tests.....	202
9.11	SubscribeCOVProperty Service Execution Tests.....	207
9.12	AtomicReadFile Service Execution Tests.....	214
9.13	AtomicWriteFile Service Execution Tests.....	220
9.14	AddListElement Service Execution Tests.....	230
9.15	RemoveListElement Service Execution Tests.....	232
9.16	CreateObject Service Execution Tests.....	234
9.17	DeleteObject Service Execution Tests.....	238
9.18	ReadProperty Service Execution Tests.....	239
9.19	ReadPropertyConditional Service Execution Tests.....	241
9.20	ReadPropertyMultiple Service Execution Tests.....	242
9.21	ReadRange Service Execution Tests.....	249
9.22	WriteProperty Service Execution Tests.....	251
9.23	WritePropertyMultiple Service Execution Tests.....	256
9.24	DeviceCommunicationControl Service Execution Test.....	264
9.25	ConfirmedPrivateTransfer Service Execution Tests.....	268
9.26	UnconfirmedPrivateTransfer Service Execution Tests.....	269
9.27	ReinitializeDevice Service Execution Tests.....	269
9.28	ConfirmedTextMessage Service Execution Tests.....	271
9.29	UnconfirmedTextMessage Service Execution Tests.....	273
9.30	TimeSynchronization Service Execution Tests.....	273
9.31	UTCTimeSynchronization Service Execution Tests.....	274
9.32	Who-Has Service Execution Tests.....	274
9.33	Who-Is Service Execution Tests.....	280
9.34	VT-Open Service Execution Tests.....	283
9.35	VT-Close Service Execution Tests.....	284
9.36	VT-Data Service Execution Tests.....	285
9.37	RequestKey Service Execution Test.....	286
9.38	Authenticate Service Execution Tests.....	288
9.39	General Testing of Service Execution.....	292
10	NETWORK LAYER PROTOCOL TESTS.....	293
10.1	Processing Application Layer Messages Originating from Remote Networks.....	293
10.2	Router Functionality Tests.....	293
10.3	Half-Router Functionality Tests.....	317
10.4	B/IP PAD Tests.....	323
10.5	Initiating Network Layer Messages.....	325
11	LOGICAL LINK LAYER PROTOCOL TESTS.....	327
11.1	UI Command and Response.....	327
11.2	XID Command and Response.....	327
11.3	TEST Command and Response.....	328
12	DATA LINK LAYER PROTOCOLS TESTS.....	329
12.1	MS/TP State Machine Tests.....	329
12.2	PTP State Machine Tests.....	381

13	SPECIAL FUNCTIONALITY TESTS.....	417
13.1	Segmentation.....	417
13.2	Time Master .....	426
13.3	Character Sets .....	427
13.4	Malformed PDUs .....	427
13.5	Slave Proxy Tests.....	428
14	BACnet/IP Functionality Tests .....	431
14.1	Non-BBMD B/IP Device .....	431
14.2	Non-BBMD B/IP device Device with a Server Application.....	433
14.3	Broadcast Distribution Table Operations .....	433
14.4	Foreign Device Table Operations (Negative Tests) .....	436
14.5	BACnet Broadcast Management (No Foreign Device Table, No Applications).....	437
14.6	Foreign Device Management .....	438
14.7	Broadcast Management (BBMD, Foreign Devices, Local Application).....	440
15	Reporting Test Results .....	443
	ANNEX A - Example EPICS (INFORMATIVE) .....	444

**iTeh STANDARD PREVIEW**  
(standards.iteh.ai)

Full standard:  
<https://standards.iteh.ai/catalog/standards/sist/5b942d7b-f0f4-48ca-aa86-dec20e20177/iso-16484-6-2009>

## Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

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

ISO 16484-6 was prepared by Technical Committee ISO/TC 205, *Building environment design*.

This second edition cancels and replaces the first edition (ISO 16484-6:2005), of which it constitutes a minor revision.

ISO 16484 consists of the following parts, under the general title *Building automation and control systems (BACS)*:

- *Part 2: Hardware*
- *Part 3: Functions*
- *Part 5: Data communication protocol*
- *Part 6: Data communication conformance testing*

A Part 1, dealing with project implementation, and a Part 4, dealing with applications, are under development.

**iTeh STANDARD PREVIEW**  
**(standards.iteh.ai)**

Full standard:  
<https://standards.iteh.ai/catalog/standards/sist/5b942d7b-f0f4-48ca-aa86-dec20e20177/iso-16484-6-2009>



# Building automation and control systems (BACS) —

## Part 6: Data communication conformance testing

### 1 Scope

This part of ISO 16484 defines a standard method for verifying that an implementation of the BACnet protocol provides each capability claimed in its Protocol Implementation Conformance Statement (PICS) in conformance with the BACnet standard.

This part of ISO 16484 provides a comprehensive set of procedures for verifying the correct implementation of each capability claimed on a BACnet PICS, including

- a) support of each claimed BACnet service, either as an initiator, executor, or both,
- b) support of each claimed BACnet object-type, including both required properties and each claimed optional property,
- c) support of the BACnet network layer protocol,
- d) support of each claimed data link option, and
- e) support of all claimed special functionality.

### 2 Relationship between this part of ISO 16484 and ANSI/ASHRE 135.1-2007

This part of ISO 16484 comprises, from Clause 4 onwards, the US standard ANSI/ASHRE 135.1-2007, *Method of Test for Conformance to BACnet*, published by the American National Standards Institute and the American Society of Heating, Refrigerating and Air-Conditioning Engineers.

### 3 Terms, definitions and abbreviated terms

For the purposes of this document, the following terms, definitions and abbreviated terms apply.

#### 3.1

##### **local network**

network to which a BACnet device is directly connected

#### 3.2

##### **remote network**

network that is accessible from a BACnet device only by passing through one or more routers

#### 3.3

##### **test database**

database of BACnet functionality and objects created by reading the contents of an EPICS

## ISO/FDIS 16484-6:2008(E)

BNF	Backus-Naur Form syntax
EPICS	electronic protocol implementation conformance statement
IUT	implementation under test
TCSL	testing and conformance scripting language
TD	testing device
TPI	text protocol information

**iTeh STANDARD PREVIEW**  
(standards.iteh.ai)

Full standard:  
<https://standards.iteh.ai/catalog/standards/sist/5b942d7b-f0f4-48ca-aa86-dec20e20177/iso-16484-6-2009>

## 4 ELECTRONIC PICS FILE FORMAT

An electronic protocol implementation conformance statement (EPICS) file contains a BACnet protocol implementation conformance statement expressed in a standardized text form. EPICS files are machine and human readable representations of the implementation of BACnet objects and services within a given device. EPICS files shall use the extension ".TPI" (text protocol information) and contain normal editable text lines consisting of text character codes ending in carriage return/linefeed pairs (X'0D', X'0A').

EPICS files are used by software testing tools to conduct and interpret the results of tests defined in this standard. An EPICS file shall accompany any device tested according to the procedures of this standard.

### 4.1 Character Encoding

BACnet provides for a variety of possible character encodings. The character encodings in BACnet fall into three groups: octet streams, double octet streams and quad octet streams. Octet streams represent characters as single octet values. In some cases, such as Microsoft DBCS and JIS C 6226, certain octet values signal that the second octet which follows should be viewed along with the leading octet as a single value, thus extending the range to greater than 256 possible characters. In contrast, double octet streams view pairs of octets as representing single characters. The ISO 10646 UCS-2 encoding is an example. The first or leading octet of the pair is the most significant part of the value. Quad octet streams, such as ISO 10646 UCS-4, treat tuples of four octets at a time as single characters with the first or leading octet being the most significant.

To accommodate the various encodings that may be used with BACnet device descriptions, EPICS files begin with a header that serves both to identify the file as an EPICS file, and to identify the particular encoding used. The header begins with the string "PICS #" where # is replaced by a numeral representing the character set as shown in Table 4-1.

**Table 4-1.** Character Set Codes

code	character set
0	ANSI X3.4
1	Microsoft DBCS
2	JIS C 6226
3	ISO 10646 (UCS-4)
4	ISO 10646 (UCS-2)
5	ISO 8859-1

An octet stream format can be recognized by examining the first eight octets of the EPICS file. Using ANSI X3.4 encoding as an example these eight octets will contain: X'50' X'49' X'43' X'53' X'20' X'30' X'0D' X'0A'. This represents the text "PICS 0" followed by carriage return and linefeed.

A double octet stream format can be recognized by examining the first 16 octets of the EPICS file. Using ISO 10646 UCS-2 encoding as an example these 16 octets will contain:

```
X'00' X'50' X'00' X'49' X'00' X'43' X'00' X'53'
X'00' X'20' X'00' X'34' X'00' X'0D' X'00' X'0A'
```

This represents the text "PICS 4" followed by carriage return and linefeed.

A quad octet stream format can be recognized by examining the first 32 octets of the EPICS file. Using ISO 10646 UCS-4 as an example these 32 octets will contain:

```
X'00' X'00' X'00' X'50' X'00' X'00' X'00' X'49'
X'00' X'00' X'00' X'43' X'00' X'00' X'00' X'53'
X'00' X'00' X'00' X'20' X'00' X'00' X'00' X'33'
X'00' X'00' X'00' X'0D' X'00' X'00' X'00' X'0A'
```

This represents the text "PICS 3" followed by carriage return and linefeed.

## 4.2 Structure of EPICS Files

EPICS files consist of text lines ending in carriage return/linefeed pairs (X'0D', X'0A') encoded as octet, double octet or quad octet streams as defined in 4.1. In the rest of this standard, the term "character" will be used to mean one symbol encoded as one, two, or four octets based on the character encoding used in the EPICS file header. For example, the character space may be encoded as X'20' or X'0020' or X'00000020'. In this standard all characters will be shown in their single octet form.

The special symbol ¶ is used in this Clause to signify the presence of a carriage return/linefeed pair (X'0D0A'). Except within character strings, the character codes tab (X'09'), space (X'20'), carriage return (X'0D') and linefeed (X'0A') shall be considered to be white space. Any sequence of 1 or more white space characters shall be equivalent to a single white space character. Except within a character string, a sequence of two dashes (X'2D') shall signify the beginning of a comment which shall end with the next carriage return/linefeed pair, i.e., the end of the line upon which the -- appears. Comments shall be considered to be white space, and may thus be inserted freely.

EPICS files shall have, as their first line following the header, the literal text:

**BACnet Protocol Implementation Conformance Statement ¶**

This text serves as a signature identifying the EPICS file format.

Lines that define the sections of the EPICS (see 4.5) and the particular implementation data for a given device follow the signature line.

The EPICS file ends with a line containing the following literal text:

**End of BACnet Protocol Implementation Conformance Statement ¶**

## 4.3 Character Strings

The occurrence of a double quote (X'22'), single quote (X'27') or accent grave (X'60') shall signify character strings. For double quotes, the end of the string shall be signified by the next occurrence of a double quote, or the end of the line. For single quote or accent grave, the end of the string shall be signified by the next occurrence of a single quote (X'27'), or the end of the line. Thus strings which need to include a single quote or accent grave as a literal character in the string shall use the double quote quoting method, while strings which need to include double quote shall use the single quote or accent grave quoting method.

## 4.4 Notational Rules for Parameter Values

Within each section, parameters may need to be expressed in one of several forms. The following rules govern the format for parameters:

- (a) key words are case insensitive so that X'41' through X'5A' are equivalent to X'61' through X'7A';
- (b) null values are shown by the string "NULL";
- (c) Boolean values are shown by the strings "T" or "TRUE" if the value is true, or "F" or "FALSE" if the value is false;
- (d) integer values are shown as strings of digits, possibly with a leading minus (-): 12345 or -111;
- (e) real values are shown with a decimal point, which may not be the first or last character: 1.23, 0.02, 1.0 but not .02;
- (f) octet strings are shown as pairs of hex digits enclosed in either single quotes (X'2D') or accent graves (X'60'), and preceded by the letter "X": X'001122';
- (g) character strings are represented as one or more characters enclosed in double, single or accent grave quotes as defined in 4.3: 'text' or 'text' or "text";
- (h) bitstrings are shown as a list, enclosed by curly brackets ({ } or X'7B' and X'7D'), of true and false values: {T,T,F} or {TRUE, TRUE, FALSE}. When the actual value of a bit does not matter, a question mark is used: {T,T,?};
- (i) enumerated values are represented as named, rather than numeric, values. Enumeration names are case insensitive so that X'41' through X'5A' are equivalent to X'61' through X'7A'. The underscore (X'5F') and dash (X'2D') are considered equivalent in enumeration names. Proprietary values are shown as a named text with no whitespace and ending in a non-negative decimal numeric. Each must start with the word "proprietary": Object\_Type, proprietary-object-type-653;
- (j) dates are represented enclosed in parenthesis: (Monday, 24-January-1998). Any "wild card" or unspecified field is shown by an asterisk (X'2A'): (Monday, \*-January-1998). The omission of day of week implies that the day is unspecified: (24-January-1998);

- (k) times are represented as hours, minutes, seconds, hundredths in the format hh:mm:ss.xx: 2:05:44.00, 16:54:59.99. Any "wild card" field is shown by an asterisk (X'2A'): 16:54:\*.\*;
- (l) object identifiers are shown enclosed by parentheses, with commas separating the object type and the instance number: (analog-input, 56). Proprietary object types replace the object type enumeration with the word "proprietary" followed by the numeric value of the object type: (proprietary 700,1);
- (m) constructed data items are represented enclosed by curly brackets ({ } or X'7B' and X'7D'), with elements separated by commas. If an element is itself a constructed value, then that element shall be enclosed in curly brackets.

#### 4.4.1 Complex Parameter Values

Some parameter values, notably property values for constructed or CHOICE types of encoded values, need to use a more complex notation to represent their values. This notation is tied to the ASN.1 encoding for those property values and may appear obscure out of context. These additional rules govern the presentation of those types of parameter values:

- (a) values which are a CHOICE of application-tagged values are represented by the value of the chosen item encoded as described in 4.4;
- (b) values which are a CHOICE of context-tagged values are represented by the context tag number enclosed in square brackets, followed by the representation of the value of the chosen item;
- (c) list values (ASN.1 "SEQUENCE OF") are represented enclosed in parenthesis, with the elements of the list separated by commas. If an element is itself a constructed value, then that element shall be enclosed in curly brackets;
- (d) array values are represented enclosed in curly brackets, with the elements of the array separated by commas. If an element is itself a constructed value, then that element shall be enclosed in curly brackets.

#### 4.4.2 Specifying Limits on Parameter Values

Some properties may have restrictions on the range or resolution of their values. In order to correctly interpret the results of tests in which the value of a property is changed using WriteProperty, WritePropertyMultiple, or AddListElement then read back using ReadProperty or ReadPropertyMultiple, it is necessary to know what these restrictions are. The test database may contain restriction statements that define these constraints. The permissible restrictions and the datatypes they apply to are:

- (a) **minimum** - the minimum value for Unsigned, Integer, Real, or Double datatypes. The earliest date for the Date datatype;
- (b) **maximum** - the maximum value for Unsigned, Integer, Real, or Double datatypes. The latest date for the Date datatype;
- (c) **resolution** - the minimum guaranteed resolution for Real and Double datatypes. The minimum time resolution in seconds for the Time datatype;
- (d) **maximum length string** - the maximum length of a CharacterString or OctetString;
- (e) **maximum length list** - the maximum number of elements guaranteed to fit in a list;
- (f) **maximum length array** - the maximum number of elements in an array;
- (g) **allowed values** - a comma-delimited list of supported enumerations for an Enumerated datatype. A comma-delimited list of object types for properties that reference an external object identifier.

Restriction statements shall be listed within pointed brackets (< and >) following the default value. If there are multiple restrictions within a single set of angle brackets, then the restrictions shall be separated by a semicolon (;). A restriction statement consists of the restriction name followed by a colon (:) followed by the restriction value or, where appropriate, a comma-delimited list of possible values.

Here are some examples of property values with restriction statements as they could appear in the test database.

```
present-value: 13.4 <minimum: 0.0; maximum: 20.0; resolution: 0.1>
description: "this is a description" <maximum length string: 30>
units: milliamperes <allowed values: milliamperes, amperes>
object-property-reference: (analog input, 12) <allowed values: analog input, analog value>
```

The Units property is a special case, because changing the units can change the value of the Present\_Value property as well as any restrictions on its value. Therefore, minimum, maximum, and resolution restrictions are only valid for the default value of the Units property.