



ADDENDA

**ANSI/ASHRAE Addendum aa to
ANSI/ASHRAE Standard 135-2010**

BACnet[®] — A Data Communication Protocol for Building Automation and Control Networks

Approved by the ASHRAE Standards Committee on June 23, 2012; by the ASHRAE Board of Directors on June 27, 2012; and by the American National Standards Institute on July 26, 2012.

This addendum was approved by a Standing Standard Project Committee (SSPC) for which the Standards Committee has established a documented program for regular publication of addenda or revisions, including procedures for timely, documented, consensus action on requests for change to any part of the standard. The change submittal form, instructions, and deadlines may be obtained in electronic form from the ASHRAE Web site (www.ashrae.org) or in paper form from the Manager of Standards.

The latest edition of an ASHRAE Standard may be purchased on the ASHRAE Web site (www.ashrae.org) or from ASHRAE Customer Service, 1791 Tullie Circle, NE, Atlanta, GA 30329-2305. E-mail: orders@ashrae.org. Fax: 404-321-5478. Telephone: 404-636-8400 (worldwide), or toll free 1-800-527-4723 (for orders in US and Canada). For reprint permission, go to www.ashrae.org/permissions.

© 2012 ASHRAE

ISSN 1041-2336



ASHRAE Standing Standard Project Committee 135
Cognizant TC: TC 1.4, Control Theory and Application
SPLS Liaison: Richard L. Hall

David Robin, Chair*	David G. Holmberg	David G. Shike
Carl Neilson, Vice-Chair	Robert L. Johnson	Ted Sunderland
Bernhard Isler, Secretary*	Stephen Karg*	William O. Swan, III
Donald P. Alexander*	Simon Lemaire	David B. Thompson*
Barry B. Bridges*	J. Damian Ljungquist*	Daniel A. Traill
Coleman L. Brumley, Jr.	James G. Luth	Stephen J. Treado*
Ernest C. Bryant	John J. Lynch	Klaus Wagner
A. J. Capowski	Brian Meyers	J. Michael Whitcomb*
Clifford H. Copass	Dana Petersen	Grant N. Wichenko*
Sharon E. Dinges*	Carl J. Ruther	Christoph Zeller
Daniel P. Giorgis	Frank Schubert	Scott Ziegenfuss

*Denotes members of voting status when the document was approved for publication

ASHRAE STANDARDS COMMITTEE 2011–2012

Carol E. Marriott, Chair	Krishnan Gowri	Janice C. Peterson
Kenneth W. Cooper, Vice-Chair	Maureen Grasso	Douglas T. Reindl
Douglass S. Abramson	Cecily M. Grzywacz	Boggarm S. Setty
Karim Amrane	Richard L. Hall	James R. Tauby
Charles S. Barnaby	Rita M. Harrold	James K. Vallort
Hoy R. Bohanon, Jr.	Adam W. Hinge	William F. Walter
Steven F. Bruning	Debra H. Kennoy	Michael W. Woodford
David R. Conover	Jay A. Kohler	Craig P. Wray
Steven J. Emmerich		Eckhard A. Groll, BOD ExO
Allan B. Fraser		Ross D. Montgomery, CO

Stephanie C. Reiniche, Manager of Standards

SPECIAL NOTE

This American National Standard (ANS) is a national voluntary consensus standard developed under the auspices of ASHRAE. *Consensus* is defined by the American National Standards Institute (ANSI), of which ASHRAE is a member and which has approved this standard as an ANS, as "substantial agreement reached by directly and materially affected interest categories. This signifies the concurrence of more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that an effort be made toward their resolution." Compliance with this standard is voluntary until and unless a legal jurisdiction makes compliance mandatory through legislation.

ASHRAE obtains consensus through participation of its national and international members, associated societies, and public review.

ASHRAE Standards are prepared by a Project Committee appointed specifically for the purpose of writing the Standard. The Project Committee Chair and Vice-Chair must be members of ASHRAE; while other committee members may or may not be ASHRAE members, all must be technically qualified in the subject area of the Standard. Every effort is made to balance the concerned interests on all Project Committees.

The Manager of Standards of ASHRAE should be contacted for:

- interpretation of the contents of this Standard,
- participation in the next review of the Standard,
- offering constructive criticism for improving the Standard, or
- permission to reprint portions of the Standard.

DISCLAIMER

ASHRAE uses its best efforts to promulgate Standards and Guidelines for the benefit of the public in light of available information and accepted industry practices. However, ASHRAE does not guarantee, certify, or assure the safety or performance of any products, components, or systems tested, installed, or operated in accordance with ASHRAE's Standards or Guidelines or that any tests conducted under its Standards or Guidelines will be nonhazardous or free from risk.

ASHRAE INDUSTRIAL ADVERTISING POLICY ON STANDARDS

ASHRAE Standards and Guidelines are established to assist industry and the public by offering a uniform method of testing for rating purposes, by suggesting safe practices in designing and installing equipment, by providing proper definitions of this equipment, and by providing other information that may serve to guide the industry. The creation of ASHRAE Standards and Guidelines is determined by the need for them, and conformance to them is completely voluntary.

In referring to this Standard or Guideline and in marking of equipment and in advertising, no claim shall be made, either stated or implied, that the product has been approved by ASHRAE.

[This foreword and the “rationale” on the following page are not part of this standard. They are merely informative and do not contain requirements necessary for conformance to the standard.]

FOREWORD

Addendum 135 aa to ANSI/ASHRAE Standard 135-2010 contains a number of changes to the current standard. These modifications are the result of change proposals made pursuant to the ASHRAE continuous maintenance procedures and of deliberations within Standing Standard Project Committee 135. The proposed changes are summarized below.

135-2010 aa -1. Add Channel Object Type

135-2010 aa -2. Add WriteGroup Service

In the following document, language to be added to existing clauses of ANSI/ASHRAE 135-2010 and Addenda is indicated through the use of *italics*, while deletions are indicated by ~~strike through~~. Where entirely new subclauses are proposed to be added, plain type is used throughout.

135-2010aa-1 Add Channel Object Type.

Rationale

There are various scenarios, such as the management of groups of lighting outputs, where it is desirable to cause the same value to be written to multiple object properties in multiple objects, initiated by the same event. Using multiple BACnet WriteProperty commands can cause skewing of command execution when more than a very small number of writes must take place. Using WritePropertyMultiple is inefficient because the value and destination object identifier and property identifier must be specified for each destination each time the write occurs. It is desirable to define the destinations ahead of time as in Command objects, but without the rigidity of the Command's predefined values and mode orientation.

[Add new Clause 12.X, p. 410]

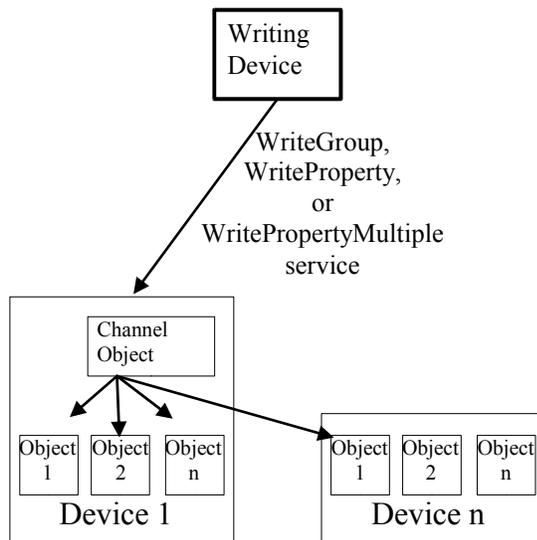
12.X Channel Object Type

The Channel object type defines a standardized object used to forward a single received value to a collection of object properties. The collection of object properties may include any combination of object types, as well as properties of different data types. The coercion of the datatype from the value written to the Channel object Present_Value to the datatypes required by the object properties is controlled by coercion rules defined in Clause 12.X.5.1.

Each Channel object is associated with a single logical “channel” in the range 0..65535. Multiple Channel object instances may be associated with a given channel number.

Each Channel object may be a member of zero or more “control groups” to facilitate writing to Channel objects with the WriteGroup service. The Channel object and its properties are summarized in Table 12-X and described in detail in this subclause.

The Channel object is intended for value distribution and does not maintain a state. Therefore, it does not act on its own and does not contain a priority array. When the Present_Value property of this object is written by the WriteProperty, WritePropertyMultiple, or WriteGroup services, and a ‘Priority’ is provided in the write, this object shall use this same priority to command the referenced properties. Figure 12-X1 illustrates the behavior of the Channel object.



Writing the same value to multiple objects using the Channel object

Figure 12-X1. Channel object behavior

When the WriteGroup service is used, potentially many devices may be affected because WriteGroup is usually broadcast. As a result, WriteGroup includes a group number parameter that restricts the effect to only those receiving devices that are members of that group. The WriteGroup further restricts the targets for writing to those Channel objects within those devices that are associated with the specified channel number(s).

Devices that contain Channel objects shall also support the WriteGroup service.

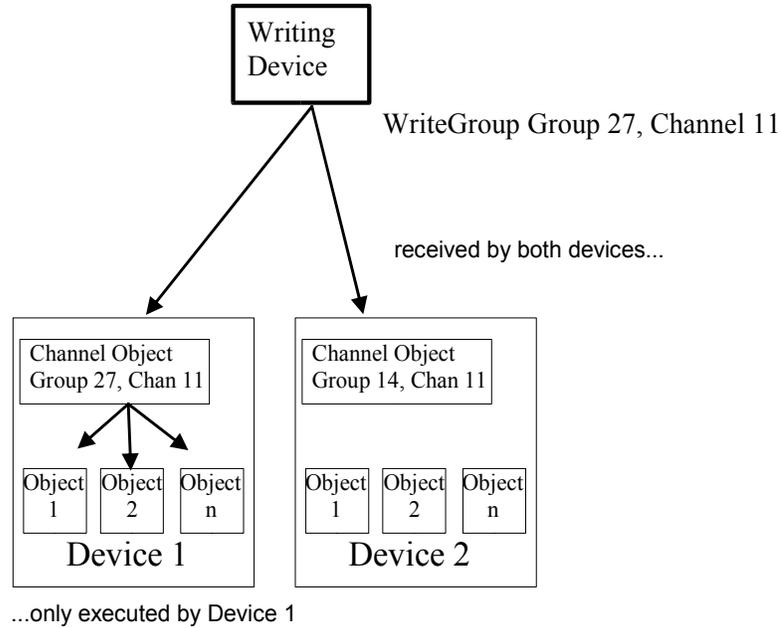


Figure 12-X2. Control Groups limit WriteGroup effect to specific Channel objects across many devices

The object and its properties are summarized in Table 12-X1 and described in detail in this subclause.

Table 12-X1. Properties of the Channel Object

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Description	CharacterString	O
Present_Value	BACnetChannelValue	W
Last_Priority	Unsigned	R
Write_Status	BACnetWriteStatus	R
Status_Flags	BACnetStatusFlags	R
Reliability	BACnetReliability	O
Out_Of_Service	BOOLEAN	R
List_Of_Object_Property_References	BACnetARRAY[N] of BACnetDeviceObjectPropertyReference	W ¹
Execution_Delay	BACnetARRAY[N] of Unsigned	O ¹
Allow_Group_Delay_Inhibit	BOOLEAN	O
Channel_Number	Unsigned16	W
Control_Groups	BACnetARRAY[N] of Unsigned32	W
Profile_Name	CharacterString	O

¹ These array properties shall be the same size.

12.X.1 Object_Identifier

This property, of type BACnetObjectIdentifier, is a numeric code that is used to identify the object. It shall be unique within the BACnet Device that maintains it.

12.X.2 Object_Name

This property, of type CharacterString, shall represent a name for the object that is unique within the BACnet Device that maintains it. The minimum length of the string shall be one character. The set of characters used in the Object_Name shall be restricted to printable characters.

12.X.3 Object_Type

This property, of type BACnetObjectType, indicates membership in a particular object type class. The value of this property shall be CHANNEL.

12.X.4 Description

This property, of type CharacterString, is a string of printable characters whose content is not restricted.

12.X.5 Present_Value (Commandable)

This property, of type BACnetChannelValue, shall indicate the value most recently written to the Present_Value.

When Present_Value is written, the Channel object shall propagate that value to each of the members in the List_Of_Object_Property_References except those members containing an empty reference. During the writing of values to members, Write_Status shall be IN_PROGRESS. At the end of writing all values, Write_Status shall change to SUCCESSFUL or FAILED based on the results of these writes. If Write_Status is SUCCESSFUL, then the Reliability property shall be reevaluated as described in 12.X.9

When Present_Value is written with a 'Priority' parameter, the resulting writes to the members of the List_Of_Object_Property_References shall also use that 'Priority' parameter. See 19.2.1.X. If the Channel object supports device-object-property references, then it may elect to use individual WriteProperty or WritePropertyMultiple, or a combination of both, to achieve the writing, as a local matter.

The initial value of the Present_Value property shall be NULL. This initial value shall not be automatically written to the properties listed in List_Of_Object_Property_References.

Attempts to write to Present_Value using WriteProperty service when Write_Status is IN_PROGRESS shall cause a Result(-) to be returned with an error class of OBJECT and an error code of BUSY.

Example List_Of_Object_Property_References [1]=(101,AV27;Present_Value)
List_Of_Object_Property_References [2]=(102,AO14;Present_Value)
List_Of_Object_Property_References [3]=(103,AO5;Present_Value)
List_Of_Object_Property_References [4]=(104,AV123;Present_Value)
Execution_Delay[1]=0
Execution_Delay[2]=100
Execution_Delay[3]=0
Execution_Delay[4]=200

t1. Present_Value written with value X
t2. If write was WriteProperty or WritePropertyMultiple, then Channel object returns Result(+) or Result(-)
t3. Write_Status = IN_PROGRESS
IF write was WriteGroup AND
WriteGroup has 'Inhibit Delay'=TRUE AND
Allow_Group_Delay_Inhibit=TRUE, THEN
{
t4. WriteProperty(101,AV27,Present_Value,X)
t5. WriteProperty(102,AO14,Present_Value,X)
t6. WriteProperty(103,AO5,Present_Value,X)
t7. WriteProperty(104,AV123,Present_Value,X)
t8. Write_Status = SUCCESSFUL
t9. Reliability = NO_FAULT_DETECTED
}
ELSE
{
t4. WriteProperty(101,AV27,Present_Value,X)
t5. WriteProperty(103,AO5,Present_Value,X)
t3+100ms. WriteProperty(102,AO14,Present_Value,X)
t3+200ms. WriteProperty(104,AV123,Present_Value,X)
t3+200ms+y Write_Status = SUCCESSFUL
t3+200ms+y Reliability = NO_FAULT_DETECTED
}

Figure 12-X3: Channel Object Execution Timeline

12.X.5.1 Datatype Coercion of Present_Value

Since List_Of_Object_Property_References can include object properties of different data types, the value written to Present_Value may require coercion to another datatype. The rules governing how these coercions occur are summarized in Table 12-X2. Those cases where Invalid Datatype (ID) is indicated in Table 12-X2, and those cases where coercion of values exceeds a range specified by an indicated coercion rule, shall be considered as coercion failures and the write shall not occur. In those cases where No Coercion (NC) is indicated in Table 12-X2, the coercion shall be considered as successful. If any of the writes to the List_Of_Object_Property_References produces a failure then Write_Status shall indicate FAILED.

Table 12-X2 – Datatype Coercion Rules

Datatype in Present_Value write	Datatype of referenced property													
	unknown	BOOLEAN	Unsigned	INTEGER	REAL	Double	OCTET STRING	CharacterString	BIT STRING	ENUMERATED	Date	Time	BACnetObjectIdentifier	BACnetLightingCommand
NULL	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	ID
BOOLEAN	NC	NC	2	2	2	2	ID	ID	ID	2	ID	ID	ID	ID
Unsigned	NC	1	NC	3	3	3	ID	ID	ID	NC	ID	ID	NC	ID
INTEGER	NC	1	4	NC	4	4	ID	ID	ID	4	ID	ID	ID	ID
REAL	NC	1	5	5	NC	5	ID	ID	ID	5	ID	ID	ID	ID
Double	NC	1	6	6	6	NC	ID	ID	ID	6	ID	ID	ID	ID
OCTET STRING	NC	ID	ID	ID	ID	ID	NC	ID	ID	ID	ID	ID	ID	ID
CharacterString	NC	ID	ID	ID	ID	ID	ID	NC	ID	ID	ID	ID	ID	ID
BIT STRING	NC	ID	ID	ID	ID	ID	ID	ID	NC	ID	ID	ID	ID	ID
ENUMERATED	NC	1	NC	3	3	3	ID	ID	ID	NC	ID	ID	ID	ID
Date	NC	ID	ID	ID	ID	ID	ID	ID	ID	ID	NC	ID	ID	ID
Time	NC	ID	ID	ID	ID	ID	ID	ID	ID	ID	ID	NC	ID	ID
BACnetObjectIdentifier	NC	ID	NC	ID	ID	ID	ID	ID	ID	ID	ID	ID	NC	ID
BACnetLightingCommand	ID	ID	ID	ID	ID	ID	ID	ID	ID	ID	ID	ID	ID	NC

NC=No Coercion ID=Invalid Datatype

12.X.5.2 Coercion Rule 1 – Numeric to BOOLEAN

The numeric value 0 maps to FALSE and anything else is TRUE.

12.X.5.3 Coercion Rule 2 – BOOLEAN to Numeric

The BOOLEAN value FALSE is mapped to 0 and TRUE is mapped to 1.

12.X.5.4 Coercion Rule 3 – Unsigned to Numeric

The Unsigned value is mapped directly to the target datatype. The Unsigned value shall be limited to 2147483647. The REAL value shall be limited in precision to seven significant digits. Values outside this limit shall cause Write_Status to indicate FAILED when the List_Of_Object_Property_References has been completely processed.

12.X.5.5 Coercion Rule 4 – INTEGER to Numeric

The INTEGER value is mapped directly to the target datatype. The Unsigned value shall be limited to 0 to 2147483647. The REAL value shall be limited in precision to seven significant digits. Values outside these limits shall cause Write_Status to indicate FAILED when the List_Of_Object_Property_References has been completely processed.

12.X.5.6 Coercion Rule 5 – REAL to Numeric

The REAL value is mapped directly to the target datatype. The Unsigned value shall be limited to 0 to 2147483000. The INTEGER value shall be limited to -2147483000 to 214783000. Values outside these limits shall cause Write_Status to indicate FAILED when the List_Of_Object_Property_References has been completely processed.

12.X.5.7 Coercion Rule 6 – Double to Numeric

The Double value is mapped directly to the target datatype. The Unsigned value shall be limited to 0 to 2147483000. The INTEGER value shall be limited to -2147483000 to 214783000. The REAL value shall be limited to $3.4 \times 10^{\pm 38}$. Values outside these limits shall cause Write_Status to indicate FAILED when the List_Of_Object_Property_References has been completely processed.

12.X.5.8 Handling of Coercion Failures

In any case of coercion failure the Write_Status shall indicate FAILED and the write shall not occur. The List_Of_Object_Property_References shall be processed in its entirety even if one or more coercion failures occur.

12.X.6 Last_Priority

This read-only property, of type Unsigned, shall convey the priority at which the Present_Value was most recently written (1..16). If an attempt was made to write to the Present_Value without the 'Priority' parameter, a default priority of 16 (the lowest priority) shall be assumed. The initial value of Last_Priority shall be 16.

12.X.7 Write_Status

This property, of type BACnetWriteStatus, shall be set to IDLE initially. This property shall be set to IN_PROGRESS when a value is written to the Present_Value property indicating that the Channel object has begun processing the List_Of_Object_Property_References.

Once all of the writes have been attempted by the Channel object, the Write_Status property shall be set to either SUCCESSFUL or FAILED. The SUCCESSFUL value indicates that the Channel object has processed all of the properties in List_Of_Object_Property_References and did not have any coercion errors, and did not receive any errors, rejects, or aborts. The FAILED value indicates that the Channel object has processed all of the properties in List_Of_Object_Property_References and encountered a coercion failure, or received an error, reject, or abort for at least one of the writes. A special exception shall be the writing of a NULL value. If a NULL value is written and WriteProperty or WritePropertyMultiple services subsequently receive an ERROR_INVALID_DATATYPE or REJECT_INVALID_PARAMETER_DATA_TYPE, it shall not be treated as a FAILED value. This is specifically to allow Channel objects to point to both commandable and non-commandable properties with the same channel.

If List_Of_Object_Property_References is empty, this property shall remain set to IDLE.

12.X.8 Status_Flags

This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of a Channel object. Two of the flags are associated with the values of another property of this object. A more detailed status could be determined by reading the property that is linked to this flag. The relationship between individual flags is not defined by the protocol. The four flags are

{IN_ALARM, FAULT, OVERRIDDEN, OUT_OF_SERVICE}

where:

IN_ALARM Logical FALSE (0).

FAULT Logical TRUE (1) if the Reliability property is present and does not have a value of NO_FAULT_DETECTED, otherwise logical FALSE (0).

OVERRIDDEN Logical TRUE (1) if the point has been overridden by some mechanism local to the BACnet Device. In this context, "overridden" is taken to mean that the Present_Value property is not changeable through BACnet services. Otherwise, the value is logical FALSE (0).

OUT_OF_SERVICE Logical TRUE (1) if the Out_Of_Service property has a value of TRUE, otherwise logical FALSE (0).

12.X.9 Reliability

This property, of type BACnetReliability, provides an indication of whether the object is "reliable" as far as the BACnet Device or operator can determine. If the Write_Status property indicates FAILED, the value of the Reliability property shall provide an indication of the type of failure that occurred. If one or more member values cannot be written because of a communication failure, the value of the Reliability property shall be COMMUNICATION_FAILURE. If one or more member values cannot be written because of invalid or inconsistent configuration, the value of the Reliability property shall be CONFIGURATION_ERROR. Other errors that may occur during the processing of writes to Present_Value shall be PROCESS_ERROR conditions. If the conditions for a PROCESS_ERROR, CONFIGURATION_ERROR, or COMMUNICATION_FAILURE are present at the same time, or some other error condition occurs, the selection of which value to use shall be a local matter.

12.X.10 Out_Of_Service

This property, of type BOOLEAN, is an indication whether (TRUE) or not (FALSE) the forwarding mechanism that the object represents is not in service. This means that changes to the Present_Value property are decoupled from the forwarding mechanism when the value of Out_Of_Service is TRUE. In addition, the Reliability property and the corresponding state of the FAULT flag of the Status_Flags property shall be decoupled from the forwarding mechanism when Out_Of_Service is TRUE. While the Out_Of_Service property is TRUE, the Present_Value and Reliability properties may still be changed to any value as a means of simulating specific fixed conditions or for testing purposes. Other functions that depend on the state of the Present_Value or Reliability properties shall respond to changes made to these properties while Out_Of_Service is TRUE, as if those changes had occurred and been passed on to the forwarding mechanism. Since the Channel object does not directly implement command prioritization, the Present_Value property shall not be required to implement the BACnet command prioritization mechanism when Out_Of_Service is TRUE. See Clause 19.

12.X.11 List_Of_Object_Property_References

This property, of type Array of BACnetDeviceObjectPropertyReference, specifies the Device Identifiers, Object Identifiers, and Property Identifiers of the properties to be written with the same value that is written to Present_Value.

This property may be restricted to only support references to objects inside of the device containing the Channel object. If the property is restricted to referencing objects within the containing device, an attempt to write a reference to an object outside the containing device into this property using WriteProperty service shall cause a Result(-) to be returned with an error class of PROPERTY and an error code of OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED.

If this property is set to reference an object outside the device containing the Channel object, the method used for writing to the referenced property value for the purpose of controlling the property is a local matter. If an implementation chooses to use WritePropertyMultiple as the preferred method of writing to the referenced property, then the device containing the Channel object shall be capable of using WriteProperty to complete writes to devices that do not support WritePropertyMultiple, or that fail before completing all required writes. If WritePropertyMultiple fails for one element, the remaining elements shall be retried as WritePropertyMultiple or WriteProperty as a local matter.

12.X.11.1 Empty References

Elements of the List_Of_Object_Property_Reference array containing object or device instance numbers equal to 4194303 are considered to be 'empty' or 'uninitialized'.

12.X.11.2 Initializing New Array Elements When the Array Size is Increased

If the size of this array is increased by writing to array index zero, each new array element shall contain an empty reference. The size of Execution_Delay shall be automatically increased to be the same.

12.X.12 Execution_Delay

This property, of type Array of Unsigned, shall indicate an execution delay in milliseconds for each value to be written in the List_Of_Object_Property_References when the Channel object's Present_Value is written. A value of zero indicates no delay. A non-zero execution delay value shall cause a delay, by that many milliseconds, in the writing to the corresponding referenced value. The resolution of Execution_Delay shall be a local matter. If present, the Execution_Delay property shall be writable. All delay periods shall "start" at the same time. So, a write of A,B(delay 100),C, D(delay 200) shall immediately

write A and C, but delay the writing of B by 100 milliseconds and D by 200 milliseconds. Multiple delayed values shall execute their corresponding delays in parallel (see Figure 12-X3).

12.X.12.1 Initializing New Array Elements When the Array Size is Increased

If the size of this array is increased by writing to array index zero, each new array element shall contain zero. The size of List_Of_Object_Property_References shall be automatically increased to be the same.

12.X.13 Allow_Group_Delay_Inhibit

This property, of type BOOLEAN, shall indicate whether WriteGroup service writes to this object, that specify 'Inhibit Delay'=TRUE, may override any execution delay specified in this object. Execution_Delay shall always occur as the result of WriteProperty or WritePropertyMultiple. In the case of WriteGroup, Execution_Delay shall always occur unless the WriteGroup service parameter 'Inhibit Delay' is TRUE, and the Channel object property Allow_Group_Delay_Inhibit is present and has the value TRUE.

12.X.14 Channel_Number

This property, of type Unsigned16, shall indicate the logical channel number that this Channel object is associated with when the Channel object Present_Value is written to using the WriteGroup service.

12.X.15 Control_Groups

This property, of type BACnetARRAY of Unsigned32, shall indicate those logical control groups of which this Channel object is a member. This array shall contain at least one entry. Unused array slots shall contain the value zero, and control group zero shall mean "no assignment." Control_Groups is required to be writable, and it shall be permitted to configure the membership of the Channel object in arbitrary groups by writing the control group numbers into this array in any order, up to the maximum number of simultaneous groups supported by the Channel object. Duplicate entries specifying the same group number shall be permitted. The maximum size of the Control_Groups array shall be a local matter.

12.X.16 Profile_Name

This property, of type CharacterString, is the name of an object profile to which this object conforms. To ensure uniqueness, a profile name must begin with a vendor identifier code (see Clause 23) in base-10 integer format, followed by a dash. All subsequent characters are administered by the organization registered with that vendor identifier code. The vendor identifier code that prefixes the profile name shall indicate the organization that publishes and maintains the profile document named by the remainder of the profile name. This vendor identifier need not have any relationship to the vendor identifier of the device within which the object resides.

A profile defines a set of additional properties, behavior, and/or requirements for this object beyond those specified here. This standard defines only the format of the names of profiles. The definition of the profiles themselves is outside the scope of this standard.

[Change **Clause 19.2.1**, p. 533]

19.2.1 Prioritization Mechanism

For BACnet objects, commands are prioritized based upon a fixed number of priorities that are assigned to command-issuing entities. A prioritized command (one that is directed at a commandable property of an object) is performed via a WriteProperty service request or a WritePropertyMultiple service request. The request primitive includes a conditional 'Priority' parameter that ranges from 1 to 16. Each commandable property of an object has an associated priority table that is represented by ~~the~~ a Priority_Array property. The Priority_Array consists of an array of commanded values in order of decreasing priority. The first value in the array corresponds to priority 1 (highest), the second value corresponds to priority 2, and so on, to the sixteenth value that corresponds to priority 16 (lowest).

...

[Change **Clause 19.2.1.1**, p. 534]

19.2.1.1 Commandable Properties

The prioritization scheme is applied to certain properties of objects. The standard commandable properties and objects are as follows:

OBJECT	COMMANDABLE PROPERTY
...	
BitString Value	Present_Value
Channel	Present_Value (see 19.2.1.X)
CharacterString Value	Present_Value
...	

The designated properties of the Analog Output, Binary Output, Multi-state Output, and Access Door objects are commandable (prioritized) by definition. The designated properties of the Analog Value, Binary Value, Multi-state Value, BitString Value, CharacterString Value, Date Value, Date Pattern Value, DateTime Value, DateTime Pattern Value, Large Analog Value, OctetString Value, Integer Value, Time Value, Time Pattern Value, and Positive Integer Value objects may optionally be commandable. Individual vendors, however, may decide to apply prioritization to any of the vendor-specified properties. These additional commandable properties shall have associated Priority_Array and Relinquish_Default properties with appropriate names. See *Clause 23.3. The Channel object is a special exception, see Clause 19.2.1.X.*

[Add new **Clause 19.2.1.X**, p. 537]

19.2.1.X Prioritization for Channel Objects

Channel objects have commandable Present_Value properties, even though the Channel object itself does not contain Priority_Array or Relinquish_Default properties. The Channel object passes the value written to Present_Value on to another object property, which may itself be commandable. In this case, any priority provided when the Channel object Present_Value is written is propagated on to its constituent member references. The Last_Priority property of the Channel object remembers the most recently provided priority value.

[Add new subclauses to **Clause K.1**, p. 855]

K.1.X1 BIBB - Data Sharing-WriteGroup-A (DS-WG-A)

The A device uses unicast, multicast or broadcast WriteGroup to target Channel object(s) in device B. The A device shall be capable of specifying any group number and any channel number.

BACnet Service	Initiate	Execute
WriteGroup	x	

The A device modifies object property values in device B by initiating WriteGroup service requests that affect Channel objects in device B.

K.1.X2 BIBB - Data Sharing-WriteGroup-Internal-B (DS-WG-I-B)

The B device shall contain one or more Channel objects that may be influenced by WriteGroup service requests from device A.

BACnet Service	Initiate	Execute
WriteGroup		x

Devices claiming conformance to DS-WG-I-B shall support configuration of Channel object BACnetDeviceObjectPropertyReference values that contain references to objects inside of device B only.

K.1.X3 BIBB - Data Sharing-WriteGroup-External-B (DS-WG-E-B)

The B device shall contain one or more Channel objects that may be influenced by WriteGroup service requests from device A.

BACnet Service	Initiate	Execute
WriteGroup		x
WriteProperty	x	

Devices claiming conformance to DS-WG-E-B shall also support DS-WG-I-B and DS-WP-A. The B device shall also support configuration of Channel object BACnetDeviceObjectPropertyReference values that contain Device Instances outside of device B, and shall be capable of initiating WriteProperty and optionally WritePropertyMultiple.

[Change **Clause 21**, **BACnetObjectType** and **BACnetObjectTypesSupported** productions, p. 609]

BACnetObjectTypes ::= ENUMERATED { -- see below for numerical order

```

...
calendar          (6),
channel           (53),
...
-- see time-value (50),
-- see channel    (53),
...
}

```

BACnetObjectTypesSupported ::= BIT STRING {

```

...
time-value        (50),
channel           (53)
}

```

[Change **Clause 21**, **BACnetPropertyIdentifier** production, p. 613]

BACnetPropertyIdentifier ::= ENUMERATED { -- see below for numerical order

```
...
all (8),
allow-group-delay-inhibit (365),
...
change-of-state-time (16),
channel-number (366),
...
configuration-files (154),
control-groups (367),
...
event-parameters (83),
execution-delay (368),
...
last-notify-record (173),
last-priority (369),
...
window-samples (148),
write-status (370),
...
-- see event-message-texts (351),
-- see allow-group-delay-inhibit (365),
-- see channel-number (366),
-- see control-groups (367),
-- see execution-delay (368),
-- see last-priority (369),
-- see write-status (370)
}
```

[Add new production to **Clause 21**, p. 588]

```
BACnetChannelValue ::= CHOICE {
null NULL,
real REAL,
enumerated ENUMERATED,
unsigned Unsigned,
boolean BOOLEAN,
signed INTEGER,
double Double,
time Time,
characterString CharacterString,
octetString OCTET STRING,
bitString BIT STRING,
date Date,
objectid BACnetObjectIdentifier,
lightingCommand [0] BACnetLightingCommand
}
```

[Add new production to **Clause 21**, p. 630]

```
BACnetWriteStatus ::= ENUMERATED {  
    idle           0,  
    in-progress   1,  
    successful     2,  
    failed        3  
}
```

[Change **Clause 21**, **BACnetPropertyStates** production, p.625]

```
BACnetPropertyStates ::= CHOICE {  
-- This production represents the possible datatypes for properties that  
-- have discrete or enumerated values. The choice must be consistent with the  
-- datatype of the property referenced in the Event Enrollment Object.  
    ...  
    backup-state          [36] BACnetBackupState,  
    write-status         [37] BACnetWriteStatus,  
    ...  
}
```

135-2010aa-2 Add WriteGroup Service.

Rationale

There are various scenarios, such as the management of groups of lighting outputs, where it is desirable to cause a large number of changes to occur, synchronized to a particular starting event. Using traditional BACnet WriteProperty commands can cause skewing of command execution when more than a very small number of writes must take place, for example, when multiple devices are involved in the same grouping because each device must be written to with a separate WriteProperty message.

[Add new **Clause 15.X**, p. 489]

15.X WriteGroup Service

The purpose of WriteGroup is to facilitate the efficient distribution of values to a large number of devices and objects. WriteGroup provides compact representations for data values that allow rapid transfer of many values. See Clause 12-X and Figure 12-X1. [referenced clauses defined in section aa-1]

The WriteGroup service is used by a sending BACnet-user to update arbitrary Channel objects' Present_Value properties for a particular numbered control group. The WriteGroup service is an unconfirmed service. Upon receipt of a WriteGroup service request, all devices that are members of the specified control group shall write to their corresponding Channel objects' Present_Value properties with the value applicable to the Channel Number, if any. A device shall be considered to be a member of a control group if that device has one or more Channel objects for which the 'Group Number' from the service appears in its Control_Groups property. If the receiving device does not contain one or more Channel objects with matching channel numbers, then those values shall be ignored.

The WriteGroup service may be unicast, multicast, broadcast locally, on a particular remote network, or using the global BACnet network address. Since global broadcasts are generally discouraged, the use of multiple directed broadcasts is preferred.

15.X.1 WriteGroup Service Structure

The structure of the WriteGroup service primitive is shown in Table 15-X. The terminology and symbology used in this table are explained in 5.6.

Table 15-X. Structure of WriteGroup Service Primitives

Parameter Name	Req	Ind
Argument	M	M(=)
Group Number	M	M(=)
Write Priority	M	M(=)
Change List	M	M(=)
Inhibit Delay	U	U(=)

15.X.1.1 Argument

The 'Argument' parameter shall convey the parameters for the WriteGroup unconfirmed service request.

15.X.1.1.1 Group Number

This parameter is an unsigned integer in the range 1 – 4,294,967,295 that represents the control group to be affected by this request. Control group zero shall never be used and shall be reserved. WriteGroup service requests containing a zero value for 'Group Number' shall be ignored.

15.X.1.1.2 Write Priority

This parameter is an unsigned integer in the range 1-16 that represents the priority for writing that shall apply to any channel value changes that result in writes to properties of BACnet objects.

15.X.1.1.3 Change List

This parameter shall specify a list of BACnetGroupChannelValue values containing at least one value. The list consists of (channel number, overridingPriority, value) tuples representing each channel number whose value is to be updated. Channel

numbers shall range from 0 to 65535 where the channel number corresponds directly to the Channel_Number property of a Channel object. The optional overridingPriority allows specific values to be written with some priority other than that specified by Write_Priority property. BACnetGroupChannelValue values convey BACnetChannelValue values that are any primitive application datatype or BACnetLightingCommand. The NULL value represents 'relinquish control' as with commandable object properties. See Clause 19.

15.X.1.1.4 Inhibit Delay

This optional parameter shall specify whether Channel objects whose Allow_Group_Delay_Inhibit properties have a value of TRUE shall inhibit any execution delay specified in their Execution_Delay property. If the 'Inhibit Delay' parameter is absent or FALSE, then execution delay(s) shall occur according to the Execution_Delay property.

15.X.2 WriteGroup Service Procedure

Since this is an unconfirmed service, no response primitives are expected. The sending BACnet-user shall transmit the WriteGroup unconfirmed request using a unicast, multicast or broadcast address. A broadcast may be sent locally, to a remote BACnet network number, or using the global BACnet network address.

If the 'Group Number' is non-zero, and the receiving BACnet-user has been configured to be a member of the control group 'Group Number' by virtue of having that group number in any of the array elements of the Control_Groups property of any of its Channel objects, then for each (channel number, overridingPriority, value) tuple provided in the 'Change List' parameter, the receiving BACnet-user shall attempt to write to the Channel object(s) whose Channel_Number property(s) match that channel number with the indicated value. If no Channel object's Channel_Number property matches the provided channel number, then that value shall be ignored.

If the optional field overridingPriority is provided, it shall specify the priority for writing the value. Otherwise the 'Write Priority' parameter shall specify the priority for writing.

If a BACnetGroupChannelValue specifies a NULL value, it shall serve the same function as if NULL had been used with WriteProperty.

The failure of any particular write shall not prevent the remaining writes from taking place.

[Change **Clause 21**, **BACnetUnconfirmedServiceChoice** production, p. 573]

```
BACnetUnconfirmedServiceChoice ::= ENUMERATED {  
    ...  
    utcTimeSynchronization      (9),  
    writeGroup                  (10)  
}
```

[Change **Clause 21**, **BACnetUnconfirmed-Service-Request** production, p. 573]

```
BACnet-Unconfirmed-Service-Request ::= CHOICE {  
    ...  
    utcTimeSynchronization [9] UTCTimeSynchronization-Request,  
    writeGroup              [10] WriteGroup-Request  
}
```

[Add new section and production to **Clause 21**, p. 575]

***** Unconfirmed Object Access Services *****

```
WriteGroup-Request ::= SEQUENCE {  
    groupNumber      [0] Unsigned32,  
    writePriority     [1] Unsigned (1..16),  
    changeList       [2] SEQUENCE of BACnetGroupChannelValue,  
    inhibitDelay     [3] BOOLEAN OPTIONAL  
}
```

[Add new production to **Clause 21**, p. 603]

```
BACnetGroupChannelValue ::= SEQUENCE {  
    channel          [0] Unsigned16,  
    overridingPriority [1] Unsigned (1..16) OPTIONAL,  
    value            BACnetChannelValue  
}
```

[Change **Clause 21**, **BACnetServicesSupported** production, p.628]

```
BACnetServicesSupported ::= BIT STRING {  
    ...  
    -- Object Access Services  
    addListElement      (8),  
    ...  
    -- writeGroup       (40),  
    writeProperty       (15),  
    ...  
    -- Services added after 1995  
    readRange           (35), -- Object Access Service  
    ...  
    getEventInformation (39) -- Alarm and Event Service  
    writeGroup          (40) -- Object Access Services  
}
```

[Add new subclauses to **Clause E.3**, p. 769]

E.3.X1 Example #1 of WriteGroup Service

We wish to set control group 23 channel 268=1111, channel 269=2222, priority for writing is 8.

```
Service = WriteGroup
'Group Number' = 23
'Write Priority' = 8
'Change List' = ((268,,1111),(269,,2222))
```

E.3.X2 Example #2 of WriteGroup Service

We wish to set control group 23 channel 12=67.0, channel 13=72.0, priority for writing is 8, inhibit execution delays.

```
Service = WriteGroup
'Group Number' = 23
'Write Priority' = 8
'Change List' = ((12,,67.0),(13,,72.0))
'Inhibit Delay' = TRUE
```

E.3.X3 Example #3 of WriteGroup Service

We wish to set control group 23 channel 12=1111 at priority 8, channel 13="ABC" at priority 10.

```
Service = WriteGroup
'Group Number' = 23
'Write Priority' = 8
'Change List' = ((12,,1111),(13,10,"ABC"))
```

[Add new subclauses to **Clause F.3**, p.794]

F.3.X Encoding for Example E.3.X1 - WriteGroup Service, Example #1

```
X'10' PDU Type=1 (BACnet-Unconfirmed-Request-PDU)
X'0A' Service Choice=10 (WriteGroup-Request)

X'09' SD Context Tag 0 (Group Number, L=1)
X'17' 23
X'19' SD Context Tag 1 (Write Priority, L=1)
X'08' 8
X'2E' PD Opening Tag 2 (Change List)
  X'0A' SD Context Tag 0 (Channel, L=2)
  X'010C' 268
  X'22' Application Tag 2 (Unsigned, L=2) (value)
  X'0457' 1111
  X'0A' SD Context Tag 0 (Channel, L=2)
  X'010D' 269
  X'22' Application Tag 2 (Unsigned, L=2) (value)
  X'08AE' 2222
X'2F' PD Closing Tag 2
```

Note that no response is required for this message since it is of type unconfirmed.

F.3.Y Encoding for Example E.3.X2 - WriteGroup Service, Example #2

X'10'	PDU Type=1 (BACnet-Unconfirmed-Request-PDU)
X'0A'	Service Choice=10 (WriteGroup-Request)
X'09'	SD Context Tag 0 (Group Number, L=1)
X'17'	23
X'19'	SD Context Tag 1 (Write Priority, L=1)
X'08'	8
X'2E'	PD Opening Tag 2 (Change List)
X'09'	SD Context Tag 0 (Channel, L=1)
X'0C'	12
X'44'	Application Tag 4 (Real, L=4) (value)
X'42860000'	67.0
X'09'	SD Context Tag 0 (Channel, L=1)
X'0D'	13
X'44'	Application Tag 4 (Real, L=4) (value)
X'42900000'	72.0
X'2F'	PD Closing Tag 2
X'39'	SD Context Tag 3 (Inhibit Delay, L=1)
X'01'	1

Note that no response is required for this message since it is of type unconfirmed.

F.3.Z Encoding for Example E.3.X3 - WriteGroup Service, Example #3

X'10'	PDU Type=1 (BACnet-Unconfirmed-Request-PDU)
X'0A'	Service Choice=10 (WriteGroup-Request)
X'09'	SD Context Tag 0 (Group Number, L=1)
X'17'	23
X'19'	SD Context Tag 1 (Write Priority, L=1)
X'08'	8
X'2E'	PD Opening Tag 2 (Change List)
X'09'	SD Context Tag 0 (Channel, L=1)
X'0C'	12
X'22'	Application Tag Unsigned L=2 (value)
X'0457'	1111
X'09'	SD Context Tag 0 (Channel, L=1)
X'0D'	13
X'19'	SD Context Tag 1 (overridingPriority, L=1)
X'0A'	10
X'74'	Application Tag Charstring L=4 (value)
X'00'	0 (Charset UTF-8)
X'414243'	"ABC"
X'2F'	PD Closing Tag 2

Note that no response is required for this message since it is of type unconfirmed.

[Add a new entry to **History of Revisions**, p. 1006]

(This History of Revisions is not part of this standard. It is merely informative and does not contain requirements necessary for conformance to the standard.)

HISTORY OF REVISIONS

...
1	14	Addendum aa to ANSI/ASHRAE 135-2010 Approved by the ASHRAE Standards Committee June 23, 2012; by the ASHRAE Board of Directors June 27, 2012; and by the American National Standards Institute July 26, 2012. 1. Add Channel Object Type 2. Add WriteGroup Service

**POLICY STATEMENT DEFINING ASHRAE'S CONCERN
FOR THE ENVIRONMENTAL IMPACT OF ITS ACTIVITIES**

ASHRAE is concerned with the impact of its members' activities on both the indoor and outdoor environment. ASHRAE's members will strive to minimize any possible deleterious effect on the indoor and outdoor environment of the systems and components in their responsibility while maximizing the beneficial effects these systems provide, consistent with accepted standards and the practical state of the art.

ASHRAE's short-range goal is to ensure that the systems and components within its scope do not impact the indoor and outdoor environment to a greater extent than specified by the standards and guidelines as established by itself and other responsible bodies.

As an ongoing goal, ASHRAE will, through its Standards Committee and extensive technical committee structure, continue to generate up-to-date standards and guidelines where appropriate and adopt, recommend, and promote those new and revised standards developed by other responsible organizations.

Through its *Handbook*, appropriate chapters will contain up-to-date standards and design considerations as the material is systematically revised.

ASHRAE will take the lead with respect to dissemination of environmental information of its primary interest and will seek out and disseminate information from other responsible organizations that is pertinent, as guides to updating standards and guidelines.

The effects of the design and selection of equipment and systems will be considered within the scope of the system's intended use and expected misuse. The disposal of hazardous materials, if any, will also be considered.

ASHRAE's primary concern for environmental impact will be at the site where equipment within ASHRAE's scope operates. However, energy source selection and the possible environmental impact due to the energy source and energy transportation will be considered where possible. Recommendations concerning energy source selection should be made by its members.

