ANSI/ASHRAE Addendum *d* to
ANSI/ASHRAE Standard 135-2001

ASHRAE® STANDARD

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

Approved by the ASHRAE Standards Committee on October 5, 2003; by the ASHRAE Board of Directors on January 29, 2004; and by the American National Standards Institute on February 25, 2004.

This standard is under continuous maintenance 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, *http://www.ashrae.org*, or in paper form from the Manager of Standards. The latest edition of an ASHRAE Standard and printed copies of a public review draft may be purchased 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 U.S. and Canada).

ISSN 1041-2336

Approved American National Standard

ANSI

AMERICAN SOCIETY OF HEATING, REFRIGERATING AND AIR-CONDITIONING ENGINEERS, INC.
1791 Tullie Circle, NE • Atlanta, GA 30329

SPECIAL NOTE

This American National Standard (ANS) is a national voluntary consensus standard developed under the auspices of the American Society of Heating, Refrigerating and Air-Conditioning Engineers (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 members may or may not be members of ASHRAE, 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:

a. interpretation of the contents of this Standard,
b. participation in the next review of the Standard,
c. offering constructive criticism for improving the Standard,
d. permission to reprint portions of the Standard.

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

FOREWORD

The purpose of this addendum is to add a number of independent substantive changes to the BACnet standard. These modifications are the result of change proposals made pursuant to the continuous maintenance procedures contained in the *Manual for Processing ASHRAE Standards* and *PC Guidance* and of deliberations within Standing Standard Project Committee 135. The changes are summarized below.

135*d*-1. Add clauses describing BACnet-EIB/KNX mapping, p. 1.

In the following document, language to be added to existing clauses of ANSI/ASHRAE 135-2001 and Addenda is indicated through the use of *italics*, while deletions are indicated by ~~strikethrough~~. Where entirely new subclauses are proposed to be added, plain type is used throughout.

135*d*-1. Add clauses describing BACnet-EIB/KNX mapping.

Addendum 135*d*-1

[Add to **Contents**, p. iv]

H.5 Using BACnet with EIB/KNX

[Add to **Clause 3.3**, p. 5]

EIB European Installation Bus
EIBA European Installation Bus Association
KNX The Konnex System Specification: EIB is the core protocol of the Konnex standard. The Konnex System Specification reflects the current status for EIB.

[Add to **Clause 21**, p. 389 ff]

BACnetEngineeringUnits ::= ENUMERATED {
…
-- *Acceleration*

meters-per-second-per-second	*(166),*

--Electrical

amperes-per-meter	*(167),*
amperes-per-square-meter	*(168),*
ampere-square-meters	*(169),*
farads	*(170),*
henrys	*(171),*
ohm-meters	*(172),*
siemens	*(173),*
siemens-per-meter	*(174),*
teslas	*(175),*
volts-per-degree-Kelvin	*(176),*
volts-per-meter	*(177),*
webers	*(178),*

--Light

candelas	*(179),*
candelas-per-square-meter	*(180),*

--Temperature

degrees-Kelvin-per-hour	*(181),*
degrees-Kelvin-per-minute	*(182),*

--Other

joule-seconds	*(183),*
radians-per-second	*(184),*
square-meters-per-Newton	*(185),*
kilograms-per-cubic-meter	*(186),*
newton-seconds	*(187),*
newtons-per-meter	*(188),*
watts-per-meter-per-degree-Kelvin	*(189),*

…

}
-- Enumerated values 0-255 are reserved for definition by ASHRAE. Enumerated values
-- 256-65535 may be used by others subject to the procedures and constraints described

-- in Clause 23. The last enumeration used in this version is ~~144~~ *189*.

[Add to **Clause 25**, p. 423]

Konnex Handbook Volume 3: System Specifications
Konnex Handbook Volume 3: System Specifications, Part 7: Interworking, Chapter 2: Datapoint Types
Konnex Handbook Volume 3: System Specifications, Part 7: Interworking, Chapter 3: Standard Identifier Tables, Annex 1 – Property Identifiers
Konnex Handbook Volume 7: Applications Descriptions

[Add to **Clause 25, Sources for Reference Material**, p. 424]

EIBA: EIB Association (EIBA) s.c.r.l., Neerveldstraat / Rue de Neerveld 105, B-1200 Brussels, Belgium
Konnex Association: Neerveldstraat / Rue de Neerveld 105, B-1200 Brussels, Belgium

[Add the following clauses, from **H.5** through **H.5.5.1**, to normative **Annex H**, p. 523]

H.5 Using BACnet with EIB/KNX

This clause describes how BACnet objects and properties are mapped to corresponding EIB/KNX Datapoints and Functional Blocks. Functional Blocks are part of the Interworking Model defined by the EIB Association / Konnex Association. In the following clauses, references to "EIB" also apply to Konnex.

H.5.1 Object Structures

The following subclauses describe the relationship between EIB Functional Blocks and BACnet objects.

H.5.1.1 EIB

EIB Functional Blocks not only describe the semantics of a function in the English language, but also define how to access the services associated with that function. This is done on the basis of "Datapoints." The Datapoints are divided into two principal categories, input and output. A Functional Block consists of a non-empty collection of one or more input and output Datapoints. At least one Datapoint is required.

Functional Blocks are contained within Physical Devices. A Physical Device implements at least one Functional Block. It may be regarded as a container for, or collection of, Functional Blocks.

H.5.1.2 BACnet

BACnet's object types define functions in terms of semantics and the services used to access these functions. To accomplish this task, BACnet object types contain properties. An object type consists of a non-empty collection of properties, some of which are mandatory while others may be optional.

BACnet also defines a Device object and a "BACnet Device" contains a collection of instances of object types. Each BACnet Device contains one, and only one, Device object. See 12.10. Typically, each physical device corresponds to a single BACnet Device and therefore contains a single Device object. An exception is described in H.2.

H.5.1.3 Relationship of EIB to BACnet

EIB Functional Blocks are comparable to BACnet object types while EIB Datapoints correlate to BACnet properties.

H.5.2 Mapping Rules for the BACnet Device Object Type

This clause provides rules for the assignment of values to the properties of the BACnet Device object type.

H.5.2.1 Object_Identifier

The Object_Identifier of a Device object is of type BACnetObjectIdentifier and must be unique internetwork-wide [OR: unique to the entire internetwork]. See 12.10.1. The encoding is defined in 20.2.14. The Object Type field (the upper 10 bits) contains the enumerated value of the BACnetObjectType. In this mapping, the content of the 22-bit Instance Number field depends on whether the Object_Identifier is identifying a Device object or some other type of object. Subject to the uniqueness constraint, a one-to-one mapping of EIB Physical Devices to BACnet Devices can be achieved by setting the upper 6 bits of the instance number to a unique EIB subnetwork identifier. This could be a part of an EIB Domain Address if it is available and is unique over all linked projects. The lower 16 bits of the instance number shall be set to the Individual Address of the EIB device. Other mapping algorithms must also ensure that the Device's BACnetObjectIdentifier remains unique internetwork wide.

H.5.2.2 Object_Name

This CharacterString is unique internetwork-wide. The string is generated from the EIB Project-Installation ID and the EIB Individual Address. The Object_Name is "EIB_Project-Installation_ID::EIB_Individual_Address". Example: For an EIB Project-Installation ID of X'0011' = D'17' and an EIB Individual Address of 1.6.7, the Object_Name string would be "17::1.6.7".

H.5.2.3 Object_Type

The value of this property shall be DEVICE (= D'8').

H.5.2.4 System_Status

Clause 12.10.4 lists the values for this BACnet property. EIB device status is determined by two variables: LoadStateMachine (LSM) and RunStateMachine (RSM). The BACnet OPERATIONAL_READ_ONLY property value is not supported by EIB devices. Table H-1 lists the relationship between EIB RunStateMachine and EIB LoadStateMachine values and BACnet System_Status.

Table H-1. Mapping of the EIB device status to BACnet System_Status

BACnet System_Status	EIB RunStateMachine	EIB LoadStateMachine
OPERATIONAL	running	Loaded
OPERATIONAL_READ_ONLY	-	-
DOWNLOAD_REQUIRED	ready	Unloaded
DOWNLOAD_IN_PROGRESS	ready	Loading
NON_OPERATIONAL	halted	Error

H.5.2.5 Vendor_Name

This CharacterString identifies the manufacturer of the EIB device, with the EIB manufacturer code in parentheses. Example: "XYZ Company (1)"

H.5.2.6 Vendor_Identifier

This is the unique Vendor Identifier code assigned by ASHRAE. If the vendor has no BACnet Vendor Identifier, this property shall be set to the Vendor Identifier for EIBA: D'74'.

H.5.2.7 Model_Name

This is the model name of the EIB device as registered with the EIBA / Konnex Association.

H.5.2.8 Firmware_Revision

The firmware revision shall be the mask version of the EIB device. The content is equal to [EIB::DeviceObject:PID_FIRMWARE_REVISION].

H.5.2.9 Application_Software_Revision

This property is the software revision of the application running on the EIB device. The content is equal to [EIB::DeviceObject:PID_PROGRAM_VERSION].

H.5.2.10 Protocol_Version

This is the version of the BACnet protocol supported by this device. See 12.10.12.

H.5.2.11 Protocol_Revision

This is the minor revision level of the BACnet protocol supported by this device. See 12.10.13.

H.5.2.12 Protocol_Services_Supported

This property indicates the BACnet protocol services supported by this device. See 12.10.14.

H.5.2.13 Protocol_Object_Types_Supported

This property indicates the BACnet protocol object types supported by this device. See 12.10.15. The protocol object types supported shall be at least Analog Input, Analog Output, Analog Value, Binary Input, Binary Output, and Binary Value.

H.5.2.14 Object_List

This property is a BACnetARRAY of Object_Identifiers. See 12.10.16.

H.5.2.15 Max_APDU_Length_Accepted

The value of this property shall be greater than or equal to 50. See 12.10.17.

H.5.2.16 Segmentation_Supported

See 12.10.18.

H.5.2.17 APDU_Timeout

See 12.10.27.

H.5.2.18 Number_Of_APDU_Retries

See 12.10.28.

H.5.2.19 Device_Address_Binding

See 12.10.33.

H.5.2.20 Database_Revision

See 12.10.34. This value shall be updated anytime the EIB device configuration is changed.

H.5.3 Mapping Rules for Other BACnet Object Types

This clause provides rules for the assignment of values to specific properties of these BACnet object types: Analog Input, Analog Output, Analog Value, Binary Input, Binary Output, and Binary Value.

H.5.3.1 Object_Identifier

An Object_Identifier is a 32-bit number that must be unique within a BACnet Device. The encoding is defined in 20.2.14. The Object Type field (the upper 10 bits) contains the enumerated value of the BACnetObjectType. In this mapping, the content of the 22-bit Instance Number field depends on whether the Object_Identifier is identifying a Device object or some other type of object. The Object_Identifiers of other object types must be unique within the BACnet device that contains them. See Clause 12. Subject to this constraint, the upper 6 bits of the instance number shall uniquely identify each instance of a mapped object contained within a BACnet Device while the remaining 16 bits shall contain the EIB Individual Address to simplify diagnostics.

H.5.3.2 Object_Name

This CharacterString is unique within the BACnet Device. The string is generated from the EIB Project-Installation ID, the EIB Individual Address, and the EIB Functional Block ID - Instance. The Object_Name is "EIB_Project-Installation_ID::EIB_Individual_Address#Functional_Block_ID-Instance". Example: For an EIB Project-Installation ID of X'0011' = D'17', an EIB Individual Address of 1.6.7, a Functional Block ID of D'10' and a Functional Block Instance of D'2' the Object_Name string would be "17::1.6.7#10-2".

H.5.3.3 Object_Type

The value of this property shall be as listed below:

ANALOG-INPUT	D'0'
ANALOG-OUTPUT	D'1'
ANALOG-VALUE	D'2'
BINARY-INPUT	D'3'
BINARY-OUTPUT	D'4'
BINARY-VALUE	D'5'

H.5.3.4 Present_Value

This property contains the present value of the Input / Output / Value. For Binary object types the permissible values are ACTIVE and INACTIVE. For Analog object types the property value is a REAL.

H.5.3.5 Description

This property contains the EIB group address associated with the Present_Value property as a CharacterString with the notation "x/y/z", where "x" is the main group, "y" is the subgroup, and "z" is the function.

H.5.3.6 Status_Flags

The values for the status flags shall be set as:

IN_ALARM	The value of this flag shall be logical FALSE (0).
FAULT	Logical TRUE (1) if the Reliability property does not have a value of NO_FAULT_DETECTED, otherwise logical FALSE (0).
OVERRIDDEN	Logical TRUE (1) if manual override can be detected by the EIB device and is executed, otherwise logical FALSE (0).

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

H.5.3.7 Event_State

The value of this property is always set to NORMAL.

H.5.3.8 Reliability

The optional "Reliability" property shall, if implemented, return at least NO_FAULT_DETECTED in case that the EIB Datapoints' Quality Codes are GOOD, and UNRELIABLE_OTHER in case at least one EIB Datapoint's Quality Code is BAD. An EIB Datapoint's Quality Code is GOOD if the Datapoint can be read from (Input) or written to (Output); otherwise its Quality Code is BAD. If an implementation is able to distinguish different sources of a failure, it may return other reliability codes of type BACnetReliability.

H.5.3.9 Out_Of_Service

The mandatory "Out_Of_Service" property shall be set to TRUE if the corresponding EIB device cannot be reached or the Present_Value cannot be read from (Input) or written to (Output) the device. Otherwise this property is set to FALSE.

H.5.3.10 Polarity

The value of this property shall always be NORMAL.

H.5.3.11 Units

The BACnet Analog Input/Output/Value Present_Value is of datatype REAL. The EIB datatypes 8-Bit Unsigned Value, 8-Bit Signed Value, 2-Octet Unsigned Value, 2-Octet Signed Value, 2-Octet Float Value, 4-Octet Unsigned Value, 4-Octet Signed Value, and 4-Octet Float Value are mapped into the datatype REAL.

These datatypes encompass a larger number of Datapoint types that are mapped into a REAL with a BACnetEngineeringUnit. See Table H-2.

Table H-2. EIB Datapoint Types

ID	Name	Range	Units	BACnetEngineeringUnits
5.001	DPT_Scaling	0...100	%	(98) percent
5.003	DPT_Angle	0...360	°	(90) degrees-angular
5.010	DPT_Value_1_Ucount	0...255	counter pulses	
6.010	DPT_Value_1_Count		counter value	
7.001	DPT_Value_2_Ucount		counter pulses (16-bit unsigned value)	
8.001	DPT_Value_2_Count		counter pulses	
9.001	DPT_Value_Temp	-273...+670760	C	(62) degrees-Celsius
9.002	DPT_Value_Tempd	-670760...+670760	K	(63) degrees-Kelvin
9.003	DPT_Value_Tempa	-670760...+670760	K/h	(181) degrees-Kelvin-per-hour
9.004	DPT_Value_Lux	0...670760	Lux	(37) luxes
9.005	DPT_Value_Wsp	0...670760	m/s	(74) meters-per-second
9.006	DPT_Value_Pres	0...670760	Pa	(53) pascals
9.010	DPT_Value_Time1	-670760...+670760	s	(73) seconds
9.011	DPT_Value_Time2	-670760...+670760	ms	(159) milliseconds
9.020	DPT_Value_Volt	-670760...+670760	mV	(124) millivolts
9.021	DPT_Value_Curr	-670760...+670760	mA	(2) milliamperes

Table H-2. EIB Datapoint Types (cont.)

ID	Name	Range	Units	BACnetEngineeringUnits
12.001	DPT_Value_4_Ucount	0...4294967295	counter pulses	
13.001	DPT_Value_4_Count	-2147483648... +2147483647	counter value	
14.000	DPT_Value_Acceleration		m s^{-2}	(166) meters-per-second-per-second
14.003	DPT_Value_Activity		s^{-1}	(101) per-second
14.005	DPT_Value_Amplitude		(unit as appropriate)	
14.006	DPT_Value_AngleRad		rad angle	(103) radians
14.007	DPT_Value_AngleDeg		° angle	(90) degrees-angular
14.008	DPT_Value_Angular_Momentum		J s	(183) joule-seconds
14.009	DPT_Value_Angular_Velocity		rad s^{-1}	(184) radians-per-second
14.010	DPT_Value_Area		m^2	(0) square-meters
14.011	DPT_Value_Capacitance		F	(170) farads
14.014	DPT_Value_Compressibility		m^2 N^{-1}	(185) square-meters-per-Newton
14.015	DPT_Value_Conductance		S = Ω^{-1}	(173) siemens
14.016	DPT_Value_ Electrical_Conductivity		S m^{-1}	(174) siemens-per-meter
14.017	DPT_Value_Density		kg m^{-3}	(186) kilograms-per-cubic-meter
14.019	DPT_Value_Electric_Current		A	(3) amperes
14.020	DPT_Value_Electric_CurrentDensity		A m^{-2}	(168) amperes-per-square-meter
14.023	DPT_Value_Electric_FieldStrength		V m^{-1}	(177) volts-per-meter
14.027	DPT_Value_Electric_Potential		V	(5) volts
14.028	DPT_Value_Electric_PotentialDifference		V	(5)volts
14.029	DPT_Value_ElectromagneticMoment		A m^2	(169) ampere-square-meters
14.030	DPT_Value_Electromotive_Force		V	(5) volts
14.031	DPT_Value_Energy		J	(16) joules
14.032	DPT_Value_Force		N	(153) newton
14.033	DPT_Value_Frequency		Hz = s^{-1}	(27) hertz
14.034	DPT_Value_Angular_Frequency		rad s^{-1}	(184) radians-per-second
14.035	DPT_Value_Heat_Capacity		J K^{-1}	(127) joules-per-degree-Kelvin
14.036	DPT_Value_Heat_FlowRate		W	(47) watts
14.037	DPT_Value_Heat_Quantity		J	(16) joules
14.038	DPT_Value_Impedance		W	(47) watts
14.039	DPT_Value_Length		m	(31) meters
14.040	DPT_Value_Light_Quantity		J or lm s	(16) joules
14.041	DPT_Value_Luminance		cd m^{-2}	(180) candelas-per-square-meter
14.042	DPT_Value_Luminous_Flux		lm	(36) lumens
14.043	DPT_Value_Luminous_Intensity		cd	(179) candelas
14.044	DPT_Value_Magnetic_FieldStrength		A m^{-1}	(167) amperes-per-meter
14.045	DPT_Value_Magnetic_Flux		Wb	(178) webers

Table H-2. EIB Datapoint Types (cont.)

ID	Name	Range	Units	BACnetEngineeringUnits
14.046	DPT_Value_Magnetic_FluxDensity		T	(175) teslas
14.047	DPT_Value_Magnetic_Moment		A m^2	(169) ampere-square-meters
14.048	DPT_Value_Magnetic_Polarization		T	(175) teslas
14.049	DPT_Value_Magnetization		A m^{-1}	(167) amperes-per-meter
14.050	DPT_Value_MagnetomotiveForce		A	(3) amperes
14.051	DPT_Value_Mass		kg	(39) kilograms
14.052	DPT_Value_MassFlux		kg s^{-1}	(42) kilograms-per-second
14.053	DPT_Value_Momentum		N s	(187) newton-seconds
14.054	DPT_Value_Phase_AngleRad		rad	(103) radians
14.055	DPT_Value_Phase_AngleDeg		°	(14) degrees-phase
14.056	DPT_Value_Power		W	(47) watts
14.057	DPT_Value_Power_Factor		cos Φ	(15) power-factor
14.058	DPT_Value_Pressure		Pa = N m^{-2}	(53) pascals
14.059	DPT_Value_Reactance		Ω	(4) ohms
14.060	DPT_Value_Resistance		Ω	(4) ohms
14.061	DPT_Value_Resistivity		Ωm	(172) ohm-meters
14.062	DPT_Value_SelfInductance		H	(171) henrys
14.064	DPT_Value_Sound_Intensity		W m^{-2}	(189) watts-per-square-meter
14.065	DPT_Value_Speed		m s^{-1}	(74) meters-per-second
14.066	DPT_Value_Stress		Pa = N m^{-2}	(53) pascals
14.067	DPT_Value_Surface_Tension		N m^{-1}	(188) newtons-per-meter
14.068	DPT_Value_Common_Temperature		C	(62) degrees-Celsius
14.069	DPT_Value_Absolute_Temperature		K	(63) degrees-Kelvin
14.070	DPT_Value_TemperatureDifference		K	(63) degrees-Kelvin
14.071	DPT_Value_Thermal_Capacity		J K^{-1}	(127) joules-per-degree-Kelvin
14.072	DPT_Value_Thermal_Conductivity		W m^{-1} K^{-1}	(190) watts-per-meter-per-degree-Kelvin
14.073	DPT_Value_ThermoelectricPower		V K^{-1}	(176) volts-per-degree-Kelvin
14.074	DPT_Value_Time		s	(73) seconds
14.075	DPT_Value_Torque		N m	(160) newton-meters
14.076	DPT_Value_Volume		m^3	(80) cubic-meters
14.077	DPT_Value_Volume_Flux		m^3 s^{-1}	(85) cubic-meters-per-second
14.078	DPT_Value_Weight		N	(153) newton
14.079	DPT_Value_Work		J	(16) joules

H.5.3.12 Priority_Array

This is an internal implementation requirement, which is not required to be mapped. See Clause 19.

H.5.3.13 Relinquish_Default

For prioritized writable properties, it is typically required by the mapping that the Present_Value shall remain unchanged when no active entry (value not equal to NULL) is present. Therefore, the Relinquish_Default shall be set equal to the Present_Value.

H.5.3.14 Profile_Name

The profile name shall be set to "74-EIB_[Profile]", where [Profile] is the name of the EIB Function Block.

H.5.4 Mappings of EIB Functional Blocks

The following Functional Block mappings are specified in this clause: Analog Input, Analog Output, Analog Value, Binary Input, Binary Output, Binary Value and Dimming Actuator.

H.5.4.1 Overview

This clause provides mappings of standardized EIB Functional Blocks to BACnet objects. These are intended to provide both standardized definitions and to serve as illustrative examples of the mapping rules prescribed above. They therefore contain additional explanations and descriptive text.

For some properties, the mapping may not be to an EIB Datapoint but rather to a static value or to a function that transforms internal information to a BACnet datatype or format.

H.5.4.2 Analog Input

The Analog Input Functional Block is mapped to the standard BACnet Analog Input object type as the semantics of these two data structures are identical and the required properties of the BACnet object type can be mapped.

Table H-3. Analog Input Mapping

Property Identifier	Property Datatype	O/R/W[1]	EIB Mapping
Object_Identifier	BACnetObjectIdentifier	R	As specified in H.5.3.1
Object_Name	CharacterString	R	As specified in H.5.3.2
Object_Type	BACnetObjectType	R	As specified in H.5.3.3
Present_Value	REAL	R	PID_ANALOG_PRESENT.Value
Description	CharacterString	O	As specified in H.5.3.5
Status_Flags	BACnetStatusFlags	R	As specified in H.5.3.6
Event_State	BACnetEventState	R	NORMAL
Reliability	BACnetReliability	O	As specified in H.5.3.8
Out_Of_Service	BOOLEAN	R	As specified in H.5.3.9
Units	BACnetEngineeringUnits	R	As specified in H.5.3.11
Min_Pres_Value	REAL	O	As specified in H.5.3.11, range lower value
Max_Pres_Value	REAL	O	As specified in H.5.3.11, range higher value
COV_Increment	REAL	O	1.0
Profile_Name	CharacterString	R	"74-EIB_AnalogInput"

[1] O/R/W = Optional, required Readable or required Writable property (see Clause 12). In the context of the mapping, a property is always required if defined as mandatory for the BACnet object type, and additional Properties which are optional in the BACnet specification or are proprietary may be defined as mandatory for the mapping. If a Property is writable, the Property must also be readable.

H.5.4.3 Analog Output

The Analog Output Functional Block is mapped to the standard BACnet Analog Output object type as the semantics of these two data structures are identical and the required properties of the BACnet object type can be mapped.

Table H-4. Analog Output Mapping

Property Identifier	Property Datatype	O/R/W	Mapping
Object_Identifier	BACnetObjectIdentifier	R	As specified in H.5.3.1
Object_Name	CharacterString	R	As specified in H.5.3.2
Object_Type	BACnetObjectType	R	As specified in H.5.3.3
Present_Value	REAL	W	PID_ANALOG_SET.Value
Description	CharacterString	O	As specified in H.5.3.5
Status_Flags	BACnetStatusFlags	R	As specified in H.5.3.6
Event_State	BACnetEventState	R	NORMAL
Reliability	BACnetReliability	O	As specified in H.5.3.8
Out_Of_Service	BOOLEAN	R	As specified in H.5.3.9
Units	BACnetEngineeringUnits	R	As specified in H.5.3.11
Priority_Array	BACnetPriorityArray	R	As specified in H.5.3.12
Relinquish_Default	REAL	R	As specified in H.5.3.13
Profile_Name	CharacterString	R	"74-EIB_AnalogOutput"

H.5.4.4 Analog Value

The Analog Value Functional Block is mapped to the standard BACnet Analog Value object type as the semantics of these two data structures are identical and the required properties of the BACnet object type can be mapped.

Table H-5. Analog Value Mapping

Property Identifier	Property Datatype	O/R/W	Mapping
Object_Identifier	BACnetObjectIdentifier	R	As specified in H.5.3.1
Object_Name	CharacterString	R	As specified in H.5.3.2
Object_Type	BACnetObjectType	R	As specified in H.5.3.3
Present_Value	REAL	R	PID_ANALOG_PRESENT.Value
Description	CharacterString	O	As specified in H.5.3.5
Status_Flags	BACnetStatusFlags	R	As specified in H.5.3.6
Event_State	BACnetEventState	R	NORMAL
Reliability	BACnetReliability	O	As specified in H.5.3.8
Out_Of_Service	BOOLEAN	R	As specified in H.5.3.9
Units	BACnetEngineeringUnits	R	As specified in H.5.3.11
COV_Increment	REAL	R	1.0
Profile_Name	CharacterString	R	"74-EIB_AnalogValue"

H.5.4.5 Binary Input

The Binary Input Functional Block is mapped to the standard BACnet Binary Input object type as the semantics of these two data structures are identical and the required properties of the BACnet object type can be mapped.

Table H-6. Binary Input Mapping

Property Identifier	Property Datatype	O/R/W	EIB Mapping
Object_Identifier	BACnetObjectIdentifier	R	As specified in H.5.3.1
Object_Name	CharacterString	R	As specified in H.5.3.2
Object_Type	BACnetObjectType	R	As specified in H.5.3.3
Present_Value	BACnetBinaryPV	R	PID_BOOLEAN_PRESENT.Value
Description	CharacterString	O	As specified in H.5.3.5
Status_Flags	BACnetStatusFlags	R	As specified in H.5.3.6
Event_State	BACnetEventState	R	NORMAL
Reliability	BACnetReliability	O	As specified in H.5.3.8
Out_Of_Service	BOOLEAN	R	As specified in H.5.3.9
Polarity	BACnetPolarity	R	NORMAL
Profile_Name	CharacterString	O	"74-EIB_BinaryInput"

H.5.4.6 Binary Output

The Binary Output Functional Block is mapped to the standard BACnet Binary Output object type as the semantics of these two data structures are identical and the required properties of the BACnet object type can be mapped.

Table H-7. Binary Output Mapping

Property Identifier	Property Datatype	O/R/W	EIB Mapping
Object_Identifier	BACnetObjectIdentifier	R	As specified in H.5.3.1
Object_Name	CharacterString	R	As specified in H.5.3.2
Object_Type	BACnetObjectType	R	As specified in H.5.3.3
Present_Value	BACnetBinaryPV	W	PID_BOOLEAN_SET.Value
Description	CharacterString	O	As specified in H.5.3.5
Status_Flags	BACnetStatusFlags	R	As specified in H.5.3.6
Event_State	BACnetEventState	R	NORMAL
Reliability	BACnetReliability	O	As specified in H.5.3.8
Out_Of_Service	BOOLEAN	R	As specified in H.5.3.9
Polarity	BACnetPolarity	R	NORMAL
Priority_Array	BACnetPriorityArray	R	As specified in H.5.3.12
Relinquish_Default	BACnetBinaryPV	R	As specified in H.5.3.13
Profile_Name	CharacterString	O	"74-EIB_BinaryOutput"

H.5.4.7 Binary Value

The Binary Value Functional Block is mapped to the standard BACnet Binary Value object type as the semantics of these two data structures are identical and the required properties of the BACnet object type can be mapped.

Table H-8. Binary Value Mapping

Property Identifier	Property Datatype	O/R/W	EIB Mapping
Object_Identifier	BACnetObjectIdentifier	R	As specified in H.5.3.1
Object_Name	CharacterString	R	As specified in H.5.3.2
Object_Type	BACnetObjectType	R	As specified in H.5.3.3
Present_Value	BACnetBinaryPV	R[1]	PID_BOOLEAN_SET.Value
Description	CharacterString	O	As specified in H.5.3.5
Status_Flags	BACnetStatusFlags	R	As specified in H.5.3.6
Event_State	BACnetEventState	R	NORMAL
Reliability	BACnetReliability	O	As specified in H.5.3.8
Out_Of_Service	BOOLEAN	R	As specified in H.5.3.9
Profile_Name	CharacterString	O	"74-EIB_BinaryValue"

[1] If Present_Value is commandable, then it is required to be writable. This property is required to be writable when Out_Of_Service is TRUE.

H.5.4.8 Dimming Actuator

The Dimming Actuator Functional Block is mapped to the standard BACnet Analog Value object, which is extended by the addition of 4 EIB-specific properties.

The present value of the EIB Dimming Actuator is always an Unsigned Integer with a range of 0 to 255 (Unsigned8). These 256 values are linearly mapped to a percentage of the maximum output, i.e., 0 = 0%, 127 = 50%, 255 = 100%. The actual physical value, if required, must be determined by the application.

The Dimming Actuator Functional Block specifies that the requested target state and current physical state be represented by different Datapoints. Although implementations may treat this internally as an identical device state (target requested and current state), this may not be the case when the dimming is the result of ramping or technically required delays. Therefore, for the two current state properties Present_Value and Present_Bin_Value, the corresponding target properties, Target_Value and Target_Bin_Value, have been defined.

Table H-9. Dimming Actuator Mapping

Property Identifier	Property Datatype	S/P[1]	O/R/W	Mapping
Object_Identifier	BACnetObjectIdentifier	S	R	As specified in H.5.3.1
Object_Name	CharacterString	S	R	As specified in H.5.3.2
Object_Type	BACnetObjectType	S	R	As specified in H.5.3.3
Description	CharacterString	S	O	As specified in H.5.3.5
Device_Type	CharacterString	S	O	Device's functional description from manufacturer data from ETS
Present_Value	REAL	S	R	PID_ANALOG_PRESENT.Value
Status_Flags	BACnetStatusFlags	S	R	As specified in H.5.3.6
Event_State	BACnetEventState	S	R	NORMAL
Reliability	BACnetReliability	S	O	As specified in H.5.3.8
Out_Of_Service	BOOLEAN	S	R	As specified in H.5.3.9
Units	BACnetEngineeringUnits	S	R	As specified in H.5.3.11
COV_Increment	REAL	S	R	1.0
Profile_Name	CharacterString	S	R	"74-EIB_DimmingActuator"
Target_Value	REAL	P	W	PID_ANALOG_SET.Value
Present_Bin_Value	BACnetBinaryPV	P	R	PID_BOOLEAN_PRESENT.Value
Target_Bin_Value	BACnetBinaryPV	P	W	PID_BOOLEAN_SET.Value
Dimming_Control[2]	REAL	P	O/W	PID_CONTROL_SET.Value

[1] S/P = Standard/Proprietary property

[2] Because BACnet specifies that a writeable property must also be readable, the value returned when the Dimming_Control property is read shall be 0.0

H.5.4.9 Defining Proprietary Object Types

If it is not possible to map an EIB Functional Block to an existing BACnet object type, a new proprietary object type must be defined. Such object types shall be assigned an object type enumeration greater than 127 and contain the Profile_Name property to uniquely identify such object types and to provide a reference to an object-specific profile or description of the object type. At a minimum, proprietary object types must have the Object_Identifier, Object_Name, Object_Type and Profile_Name properties. See Clause 23.

H.5.4.10 Defining Proprietary Properties

If it is possible to map a Functional Block to a standard BACnet object type but Datapoints exist within the EIB standard that cannot be mapped to existing object properties, or if a completely new proprietary object type with proprietary properties is defined, it will be necessary to define proprietary properties for that object type. Proprietary properties shall be assigned property identifiers greater than 511 and the profile pointed to by the Profile_Name property of the object type shall provide the property name, datatype and conformance code for each such profile-specific property. See Clause 23.

H.5.5 Additional Information

This clause provides information on the EIB Functional Blocks Specification.

H.5.5.1 EIB Functional Blocks (FBs)

Within the Functional Block specifications, the Datapoints of a device are called properties and receive a property ID in terms of a unique name within the description of a functional block. The names of the properties always start with "PID_", as in the example PID_BOOLEAN_PRESENT.

Each such Datapoint property provides the following state information at run time:

Value: The current value of the property; it may be any kind of Datapoint Type.

Timestamp: Optional; it may be absolute or relative, depending on the device's capabilities.

Qualitycode: Either GOOD or BAD.

[Add a new entry to **History of Revisions**, p.557]

(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

Protocol		***Summary of Changes to the Standard***
Version	***Revision***	
…	…	…
1	4	**Addendum *a* to ANSI/ASHRAE 135-2001** Approved by the ASHRAE Standards Committee January XX, 2004 and by the ASHRAE Board of Directors January XX, 2004; and by the American National Standards Institute XXX, 2004. 1. Add Partial Day Scheduling to the Schedule object. 2. Enable reporting of proprietary events by the Event Enrollment object. 3. Allow detailed error reporting when all ReadPropertyMultiple accesses fail. 4. Remove the Recipient property from the Event Enrollment object. 5. Add the capability to issue I-Am responses on behalf of MS/TP slave

		devices. 6. Add a new silenced mode to the DeviceCommunicationControl service. 7. Add 21 new engineering units. 8. Specify the behavior of a BACnetArray when its size is changed. 9. Clarify the behavior of a BACnet router when it receives an unknown network message type.
1	4	**Addendum *c* to ANSI/ASHRAE 135-2001** Approved by the ASHRAE Standards Committee January XX, 2004 and by the ASHRAE Board of Directors January XX, 2004; and by the American National Standards Institute XXX, 2004. 1. Allow Life Safety objects to advertise supported mode. 2. Add Unsilence Options to the LifeSafetyOperation Service. 3. Specify the relationship between the Event_Type and Event_Parameter properties. 4. Add a new Accumulator Object Type. 5. Add a new Pulse Converter Object Type. 6. Standardize event notification priorities. 7. Define Abort reason when insufficient segments are available. 8. Add new Error Codes and specify usage.
1	4	**Addendum *d* to ANSI/ASHRAE 135-2001** Approved by the ASHRAE Standards Committee January XX, 2004 and by the ASHRAE Board of Directors January XX, 2004; and by the American National Standards Institute XXX, 2004. 1. Add clauses describing BACnet-EIB/KNX mapping.

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.