



ADDENDA

**ANSI/ASHRAE Addendum cc to
ANSI/ASHRAE Standard 135-2020**



A Data Communication Protocol for Building Automation and Control Networks

Approved by ASHRAE and the American National Standards Institute on January 21, 2022.

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. Instructions for how to submit a change can be found on the ASHRAE® website (<https://www.ashrae.org/continuous-maintenance>).

The latest edition of an ASHRAE Standard may be purchased on the ASHRAE website (www.ashrae.org) or from ASHRAE Customer Service, 180 Technology Parkway NW, Peachtree Corners, GA 30092. E-mail: orders@ashrae.org. Fax: 678-539-2129. 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.

© 2022 ASHRAE

ISSN 1041-2336



ASHRAE Standing Standard Project Committee 135

Cognizant TC: 1.4, Control Theory and Application

SPLS Liaison: Charles S Barnaby

Coleman L. Brumley, Jr.*, <i>Chair</i>	Bernhard Isler	Frank Schubert
Scott Ziegenfus, <i>Vice-Chair</i>	Jake Kopocis*	Marcelo Richter da Silva
Nate Benes*	Thomas Kurowski*	Ted Sunderland
Steven T Bushby*	Shahid Naeem	Lori Tribble
Salvatore Cataldi*	Frank V. Neher*	James F. Butler
Tyler Cove	Carl Neilson*	David Fisher
Brandon M. DuPrey*	Michael Osborne	Grant N. Wichenko*
Siddharth Goyal	Scott Reed	
Alexander Gurciullo*	David Robin*	

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

ASHRAE STANDARDS COMMITTEE 2021–2022

Rick M. Heiden, <i>Chair</i>	Essam E. Khalil	Justin M. Prosser
Susanna S. Hanson, <i>Vice-Chair</i>	Malcolm D. Knight	David Robin
Charles S. Barnaby	Jay A. Kohler	Lawrence J. Schoen
Robert B. Burkhead	Cesar L. Lim	Steven C. Sill
Thomas E. Cappellin	Paul A. Lindahl, Jr.	Christian R. Taber
Douglas D. Fick	James D. Lutz	Russell C. Tharp
Michael W. Gallagher	Julie Majurin	William F. Walter
Patricia Graef	Lawrence C. Markel	Craig P. Wray
Srinivas Katipamula	Margret M. Mathison	Jaap Hogeling, BOD ExO
Gerald J. Kettler	Gwelen Paliaga	Tim J. McGinn, CO

Connor Barbaree, *Senior 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 Senior 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, the table of contents, the introduction, and the “rationales” on the following pages are not part of this standard. They are merely informative and do not contain requirements necessary for conformance to the standard.]

FOREWORD

The purpose of this addendum is to present a proposed change for public review. 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.

135-2020cc-1. Update the Network Port Object and Add BACnet/SC Configuration Support, p. 3.

135-2020cc-2. Modifications to Annex AB, p. 38.

135-2020cc-3. Add a Procedure to Replace BACnet/SC Certificates, p. 40.

135-2020cc-4. Add Network Port Object Configuration BIBBs, p.44.

In the following document, language to be added to existing clauses of ANSI/ASHRAE 135-2020 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. Only this new and deleted text is open to comment at this time. All other material in this document is provided for context only and is not open for public review comment except as it relates to the proposed changes.

The use of placeholders like XX, YY, ZZ, X1, X2, NN, x, n, ? etc. should not be interpreted as literal values of the final published version. These placeholders will be assigned actual numbers/letters only after final publication approval of the addendum.

135-2020cc-1. Update the Network Port Object and Add BACnet/SC Configuration Support

Rationale

135-2020 Clause AB (BACnet/SC) requires configuration items, status indications, and other elements to be network visible and interoperably configurable.

The Network Port object is extended to allow representation and configuration of BACnet/SC.

[Network Port Object Property Table Changes]

[Change **Clause 12.56**, p. 542]

12.56 Network Port Object Type

The Network Port object provides access to the configuration and properties of network ports of a device. All BACnet devices shall contain at least one Network Port object per physical port which the device can be configured to communicate BACnet through (unless the port is currently for communications on a network other than the current BACnet internetwork and this use precludes its use for the current BACnet internetwork). It is a local matter whether or not Network Port objects exist for non-configured ports. It is a local matter whether or not the Network Port object is used for non-BACnet ports.

Verification and validation of property values within a Network Port object is a local matter.

Property values which are required to maintain proper operation of the network shall be retained across a device reset.

~~The Network Port object type can be implemented as a single interface through which all of the settings for a network port are accessed, or the Network Port objects can be organized in a hierarchy which separates the settings for each communication protocol level. See Clause 12.56.10 for more details on hierarchical Network Port objects.~~

Network Port objects may optionally support intrinsic reporting to facilitate the reporting of fault conditions. Network Port objects that support intrinsic reporting shall apply the NONE event algorithm.

~~As specified in Table 12-71 and the text below, some properties of the Network Port object are required if the object is used to represent a network of a given type. For example, a Network Port object whose Network_Type is MSTP and the node is an MS/TP master node shall include the Max_Master property, and a Network Port object whose Network_Type is IPV4 shall include the IP_Subnet_Mask property. Aside from the properties so required, it is a local matter whether a Network Port object contains properties that do not apply to its Network_Type. For example, a Network Port object whose Network_Type is MSTP may include the IP_Subnet_Mask property, although the value of this property would not be used by the network. Some vendors may find it convenient to have all of their Network Port objects support the same list of properties regardless of Network_Type. This is permitted, but not required.~~

The Network Port object type can be implemented as a single interface through which all of the settings for a network port are accessed, or Network Port objects can be organized in a hierarchy which separates the settings for each communication protocol level.

Hierarchical Network Port objects

Support for hierarchical Network Port objects is optional.

In most cases, a single hierarchy chain consists of a Network Port object with a Protocol_Level of PHYSICAL at the bottom, one or more Network Port objects with their Protocol_Level set to PROTOCOL, and a Network Port object with a Protocol_Level of BACNET_APPLICATION at the top.

In some cases, a higher-level Network Port object may reference more than one lower-level Network Port object. In that case, the Reference_Port property shall contain one of the referenced Network Port objects and the Additional_Reference_Ports property shall contain the remaining referenced Network Port objects.

If lower protocol levels are shared, then multiple BACNET_APPLICATION Network Port objects can reference the same PROTOCOL or PHYSICAL Network Port object and multiple PROTOCOL Network Port objects can reference the same PHYSICAL Network Port object.

A Network Port object with a Protocol_Level of BACNET_APPLICATION or PHYSICAL shall not be in the middle of a hierarchy chain.

A hierarchical Network Port object shall contain all required properties in Table 12-71 and the required properties in the sub-table that represents the Network_Type and the column that represents the Protocol_Level for the Network Port object. The Network Port object shall not contain any standard properties other than the required and optional properties specified in Table 12.71 and the sub-tables that represent the Network_Type and the column that represents the Protocol_Level.

Non-hierarchical Network Port objects

A non-hierarchical Network Port object contains the same properties as the equivalent chain of hierarchical Network Port objects but rolled up into a single Network Port object. Using a non-hierarchical Network Port object is not allowed when a roll up of the equivalent chain of hierarchical Network Port objects would result in a conflict between the same-named properties at the same protocol level.

The Protocol_Level property shall be BACNET_APPLICATION or NON_BACNET_APPLICATION and the Reference_Port and Additional_Reference_Ports properties shall be absent.

Where the same-named property is specified in multiple Network Port objects in the equivalent hierarchy chain at different protocol levels, the property's value in the Network Port object nearest to the top of the chain shall be the property reflected in a non-hierarchical Network Port object.

BACnet MS/TP Example

A hierarchy of Network Port objects for a BACnet MS/TP port will be a chain of three Network Port objects. See Figure 12-X1. The highest level will be a Network Port object with Network_Type of MSTP at Protocol_Level of BACNET_APPLICATION that will contain all the required properties from the BACNET_APPLICATION column of Table 12-71.6. The next level will be a Network Port object with Network_Type of MSTP at Protocol_Level of PROTOCOL that will contain all the required properties from the PROTOCOL column of Table 12-71.6. The lowest level will be a Network Port object with Network_Type of SERIAL at Protocol_Level of PHYSICAL that will contain all the required properties from the PHYSICAL column of Table 12-71.9.

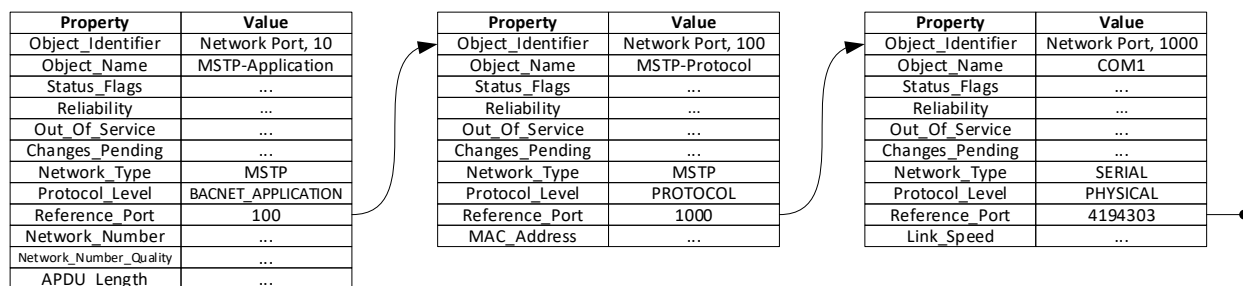


Figure 12-X1. Example of a Network Port Hierarchy Chain for BACnet MS/TP

Figure 12-X2 is an example of a single non-hierarchical Network Port object that represents a BACnet MS/TP.

Property	Value
Object_Identifier	Network Port, 10
Object_Name	MSTP-Application
Status_Flags	...
Reliability	...
Out_Of_Service	...
Changes_Pending	...
Network_Type	MSTP
Protocol_Level	BACNET_APPLICATION
Network_Number	...
Network_Number_Quality	...
APDU_Length	...
MAC_Address	...
Link_Speed	...

Figure 12-X2. Example of a Single Non-Hierarchical Network Port object for BACnet MS/TP port

BACnet/IPv6 over MS/TP Example

A hierarchy of Network Port objects for a BACnet/IPv6 port that uses MS/TP as the transport will require a chain of four Network Port objects. See Figure 12-X3. The highest level will be a Network Port object with Network_Type of IPV6 and Protocol_Level of BACNET_APPLICATION and will contain all the required properties from the BACNET_APPLICATION column of Table 12-71.4. The next level will be a Network Port object with Network_Type of IPV6 and Protocol_Level of PROTOCOL and will contain all the required properties from the PROTOCOL column of Table 12-71.4. The next level will be a Network Port object with Network_Type of MSTP and Protocol_Level of PROTOCOL and will contain all the required properties from the PROTOCOL column of Table 12-71.6. The lowest level will be a Network Port object with Network_Type of SERIAL and Protocol_Level of PHYSICAL and will contain all the required properties from the PHYSICAL column of Table 12-71.9.

Property	Value
Object_Identifier	Network Port, 1
Object_Name	IPv6-Application
Status_Flags	...
Reliability	...
Out_Of_Service	...
Changes_Pending	...
Network_Type	IPv6
Protocol_Level	BACNET_APPLICATION
Reference_Port	256
Network_Number	...
Network_Number_Quality	...
APDU_Length	...
MAC_Address	...
BACnet_IPv6_Mode	...
BACnet_IPv6_UDP_Port	...
BACnet_IPv6_Multicast_Address	...
Virtual_MAC_Address_Table	...

Property	Value
Object_Identifier	Network Port, 256
Object_Name	IPv6-Protocol
Status_Flags	...
Reliability	...
Out_Of_Service	...
Changes_Pending	...
Network_Type	IPv6
Protocol_Level	PROTOCOL
Reference_Port	257
IPv6_Address	...
IPv6_Prefix_Length	...
IPv6_Default_Gateway	...
IPv6_DNS_Server	...

Property	Value
Object_Identifier	Network Port, 257
Object_Name	MSTP-Protocol
Status_Flags	...
Reliability	...
Out_Of_Service	...
Changes_Pending	...
Network_Type	MSTP
Protocol_Level	PROTOCOL
Reference_Port	258
MAC_Address	...

Property	Value
Object_Identifier	Network Port, 258
Object_Name	COM1
Status_Flags	...
Reliability	...
Out_Of_Service	...
Changes_Pending	...
Network_Type	SERIAL
Protocol_Level	PHYSICAL
Reference_Port	4194303
Link_Speed	...

Figure 12-X3. Example of a Network Port Hierarchy Chain for BACnet/IPv6 over MS/TP

Figure 12-X4 is an example of a BACnet/IPv6 over MS/TP port represented by a single non-hierarchical Network Port object. Where a property is specified in multiple Network Port objects in the equivalent hierarchy chain, the property's value in the Network Port object nearest to the top of the chain shall be the property reflected in a non-hierarchical Network Port object.

For example, the *MAC_Address* property in Figure 12-X4 contains the IPv6 VMAC_address from the BACNET_APPLICATION protocol level and not the 1-octet MS/TP MAC address from the MSTP protocol level.

Property	Value
Object_Identifier	Network Port, 1
Object_Name	IPv6-Application
Status_Flags	...
Reliability	...
Out_Of_Service	...
Changes_Pending	...
Network_Type	IPv6
Protocol_Level	BACNET_APPLICATION
Network_Number	...
Network_Number_Quality	...
APDU_Length	...
MAC_Address	...
BACnet_IPv6_Mode	...
BACnet_IPv6_UDP_Port	...
BACnet_IPv6_Multicast_Address	...
Virtual_MAC_Address_Table	...
IPv6_Address	...
IPv6_Prefix_Length	...
IPv6_Default_Gateway	...
IPv6_DNS_Server	...
Link_Speed	...

Figure 12-X4. Example of a Single Network Port object for BACnet/IPv6 over MS/TP

Table 12.71 contains the required and optional properties common to all Network Port objects independent of the *Network_Type* and *Protocol_Level*.

Table 12-71. Base Properties of the Network Port Object Type

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Description	CharacterString	O
Status_Flags	BACnetStatusFlags	R
Reliability	BACnetReliability	R
Out_Of_Service	BOOLEAN	R
Network_Type	BACnetNetworkType	R
Protocol_Level	BACnetProtocolLevel	R
Reference_Port	Unsigned	O ¹
Additional_Reference_Ports	BACnetList of Unsigned	O²
Network_Number	Unsigned16	O
Network_Number_Quality	BACnetNetworkNumberQuality	O
Changes_Pending	BOOLEAN	R
Command	BACnetNetworkPortCommand	O ²³
Current_Health	BACnetPortHealth	O
Command_Validation_Result	BACnetPortHealth	O
MAC_Address	OCTET STRING	O ³
APDU_Length	Unsigned	R O
Link_Speed	REAL	R
Link_Speeds	BACnetARRAY[N] of REAL	O⁴
Link_Speed_Autonegotiate	BOOLEAN	O
Network_Interface_Name	CharacterString	O
BACnet_IP_Mode	BACnetIPMode	O ⁵
IP_Address	OCTET STRING	O ⁶
BACnet_IP_UDP_Port	Unsigned16	O ⁵
IP_Subnet_Mask	OCTET STRING	O ⁶
IP_Default_Gateway	OCTET STRING	O ⁶
BACnet_IP_Multicast_Address	OCTET STRING	O ⁷
IP_DNS_Server	BACnetARRAY[N] of OCTET STRING	O ⁶
IP_DHCP_Enable	BOOLEAN	O ⁸

IP_DHCP_Lease_Time	Unsigned	Q
IP_DHCP_Lease_Time_Remaining	Unsigned	Q
IP_DHCP_Server	OCTET STRING	Q
BACnet_IP_NAT_Traversal	BOOLEAN	Q ⁹
BACnet_IP_Global_Address	BACnetHostNPort	Q ¹⁰
BBMD_Broadcast_Distribution_Table	BACnetLIST of BACnetBDTEntry	Q ¹¹
BBMD_Accept_FD_Registrations	BOOLEAN	Q ¹¹
BBMD_Foreign_Device_Table	BACnetLIST of BACnetFDTEEntry	Q ¹²
FD_BBMD_Address	BACnetHostNPort	Q ¹³
FD_Subscription_Lifetime	Unsigned16	Q ¹³
BACnet_IPv6_Mode	BACnetIPMode	Q ¹⁴
IPv6_Address	OCTET STRING	Q ¹⁵
IPv6_Prefix_Length	Unsigned8	Q ¹⁵
BACnet_IPv6_UDP_Port	Unsigned16	Q ¹⁴
IPv6_Default_Gateway	OCTET STRING	Q ¹⁵
BACnet_IPv6_Multicast_Address	OCTET STRING	Q ¹⁴
IPv6_DNS_Server	BACnetARRAY[N] of OCTET STRING	Q ¹⁵
IPv6_Auto_Addressing_Enable	BOOLEAN	Q ¹⁶
IPv6_DHCP_Lease_Time	Unsigned	Q
IPv6_DHCP_Lease_Time_Remaining	Unsigned	Q
IPv6_DHCP_Server	OCTET STRING	Q
IPv6_Zone_Index	CharacterString	Q ¹⁷
Max_Master	Unsigned8	Q ¹⁸
Max_Info_Frames	Unsigned8	Q ¹⁸
Slave_Proxy_Enable	BOOLEAN	Q ¹⁹
Manual_Slave_Address_Binding	BACnetLIST of BACnetAddressBinding	Q ¹⁹
Auto_Slave_Discovery	BOOLEAN	Q ²⁰
Slave_Address_Binding	BACnetLIST of BACnetAddressBinding	Q ²¹
Virtual_MAC_Address_Table	BACnetLIST of BACnetVMACEntry	Q ²²
Routing_Table	BACnetLIST of BACnetRouterEntry	Q
Event_Detection_Enable	BOOLEAN	Q ^{23,24,5}
Notification_Class	Unsigned	Q ^{23,24,5}
Event_Enable	BACnetEventTransitionBits	Q ^{23,24,5}
Acked_Transitions	BACnetEventTransitionBits	Q ^{23,24,5}
Notify_Type	BACnetNotifyType	Q ^{23,24,5}
Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	Q ^{23,24,5}
Event_Message_Texts	BACnetARRAY[3] of CharacterString	Q ²⁴⁵
Event_Message_Texts_Config	BACnetARRAY[3] of CharacterString	Q ²⁴⁵
Event_State	BACnetEventState	Q ²³⁴
Reliability_Evaluation_Inhibit	BOOLEAN	O
Property_List	BACnetARRAY[N] of BACnetPropertyIdentifier	R
Tags	BACnetARRAY[N] of BACnetNameValue	O
Profile_Location	CharacterString	O
Profile_Name	CharacterString	O

[†] Required to be writable in routers and any other device that requires knowledge of the network number for proper operation.

¹ Required to be present if, and only if, this port can be part of a hierarchy.

² Shall be present only if the Reference_Port property is present.

²³ Required to be present if, and only if, the object supports execution of any of the values of the Command property. If present, this property shall be writable.

³ Required if the port is not a PTP link. Read only if the port is a BACnet/IP port or if the network represented by this object requires VMAC addressing.

⁴ Required if Link_Speed is writable.

⁵ Required to be present if the port is a BACnet/IP port.

⁶ Required if the port is a BACnet/IP port. If the BACnet_IP_DHCP property is TRUE, and this property is configured by DHCP, this property shall be read only.

⁷ Required to be present if Network_Type is IPV4, Protocol_Level is BACNET_APPLICATION and the port supports multicast.

- ⁸ ~~Shall be present if, and only if, Network_Type is IPV4 and the port can be configured by DHCP.~~
- ⁹ ~~Required to be present if the Network_Type is IPV4, Protocol_Level is BACNET_APPLICATION, and the is capable of communicating through a NAT router as described in Clause J.7.8.~~
- ¹⁰ ~~Required if Network_Type is IPV4, Protocol_Level is BACNET_APPLICATION, and the device is configured to communicate through a NAT router as described in Clause J.7.8.~~
- ¹¹ ~~Required to be present if Network_Type is IPV4 or IPV6 and the device is capable of functioning as a BBMD.~~
- ¹² ~~Required if Network_Type is IPV4 or IPV6 and the device is capable of functioning as a BBMD.~~
- ¹³ ~~Required to be present if Network_Type is IPV4 or IPV6 and BACnet_IP_Mode or BACnet_IPv6_Mode respectively is set to FOREIGN.~~
- ¹⁴ ~~Required to be present if Network_Type is IPV6 and Protocol_Level is BACNET_APPLICATION.~~
- ¹⁵ ~~Required to be present if Network_Type is IPV6 . Read only if the value is configured by automatic address assignment.~~
- ¹⁶ ~~Required to be present if Network_Type is IPV6 and the port supports automatic IPv6 address assignment.~~
- ¹⁷ ~~Required to be present if Network_Type is IPV6 and the node supports multiple IPv6 link local addresses.~~
- ¹⁸ ~~Required to be present and writable if Network_Type is MSTP, the device is an MS/TP master node, and the device supports the WriteProperty service.~~
- ¹⁹ ~~Required to be present and writable if Network_Type is MSTP, and the device is capable of being a Slave Proxy device.~~
- ²⁰ ~~Required if Network_Type is MSTP, Protocol_Level is BACNET_APPLICATION, and the device is capable of being a Slave Proxy device that implements automatic discovery of slaves.~~
- ²¹ ~~Required if Network_Type is MSTP, Protocol_Level is BACNET_APPLICATION, and the device is capable of being a Slave Proxy device.~~
- ²² ~~Required if the network represented by this object requires VMAC addressing.~~
- ⁴²³ ~~These properties are required if the object supports intrinsic reporting.~~
- ⁵²⁴ ~~These properties shall be present only if the object supports intrinsic reporting.~~

The below 12-71 sub-tables provide the information needed to determine what properties are allowed to be included in each hierarchical Network Port object and whether each property is required or optional. To clearly identify these properties, each sub-table represents the allowed properties for a specific Network_Type and each sub-table is further broken down by Protocol_Level.

When processing the special property identifiers ALL, REQUIRED, and OPTIONAL, all the properties in the below sub-tables are treated as conformance code **O**.

The properties that are required based on the Network_Type and Protocol_Level are identified in the sub-tables with a **(R)** conformance code. The properties that are optional based on the Network_Type and Protocol_Level are identified in the sub-tables with an **O** conformance code.

Table 12-71.1. Additional Properties of the Network Port Object Type if Network_Type is ARCNET

Property Identifier	Property Datatype	BACNET_APPLICATION	PROTOCOL	PHYSICAL
Network_Number	Unsigned16	(R)		
Network_Number_Quality	BACnetNetworkNumberQuality	(R)		
APDU_Length	Unsigned	(R)		
Routing_Table	BACnetLIST of BACnetRouterEntry	O		
MAC_Address	OCTET STRING			(R) ¹
Link_Speed	REAL			O
Link_Speeds	BACnetARRAY[N] of REAL			O
Link_Speed_Autonegotiate	BOOLEAN			O
Network_Interface_Name	CharacterString			O

¹ Required to be configurable.

Table 12-71.2. Additional Properties of the Network Port Object Type if Network Type is ETHERNET

Property Identifier	Property Datatype	BACNET_APPLICATION	PROTOCOL	PHYSICAL
Network_Number	Unsigned16	(R)		
Network_Number_Quality	BACnetNetworkNumberQuality	(R)		
APDU_Length	Unsigned	(R)		
Routing_Table	BACnetLIST of BACnetRouterEntry	O		
MAC_Address	OCTET STRING			(R) ¹
Link_Speed	REAL			O
Link_Speeds	BACnetARRAY[N] of REAL			O
Link_Speed_Autonegotiate	BOOLEAN			O
Network_Interface_Name	CharacterString			O

¹ Shall contain the six octet Ethernet MAC address.

Table 12-71.3. Additional Properties of the Network Port Object Type if Network Type is IPV4

Property Identifier	Property Datatype	BACNET_APPLICATION	PROTOCOL	PHYSICAL
Network_Number	Unsigned16	(R)		
Network_Number_Quality	BACnetNetworkNumberQuality	(R)		
APDU_Length	Unsigned	(R)		
MAC_Address	OCTET STRING	(R) ¹		
BACnet_IP_Mode	BACnetIPMode	(R)		
BACnet_IP_UDP_Port	Unsigned16	(R)		
Routing_Table	BACnetLIST of BACnetRouterEntry	O		
BACnet_IP_Multicast_Address	OCTET STRING	O ²		
BACnet_IP_NAT_Traversal	BOOLEAN	O ³		
BACnet_IP_Global_Address	BACnetHostNPort	O ³		
BBMD_Broadcast_Distribution_Table	BACnetLIST of BACnetBDTEntry	O ⁴		
BBMD_Accept_FD_Registrations	BOOLEAN	O ⁴		
BBMD_Foreign_Device_Table	BACnetLIST of BACnetFDTEntry	O ⁴		
FD_BBMD_Address	BACnetHostNPort	O ⁵		
FD_Subscription_Lifetime	Unsigned16	O ⁵		
IP_Address	OCTET STRING		(R) ⁶	
IP_Subnet_Mask	OCTET STRING		(R) ⁶	
IP_Default_Gateway	OCTET STRING		(R) ⁶	
IP_DNS_Server	BACnetARRAY[N] of OCTET STRING		(R) ⁶	
IP_DHCP_Enable	BOOLEAN		O ⁷	
IP_DHCP_Lease_Time	Unsigned		O ⁸	
IP_DHCP_Lease_Time_Remaining	Unsigned		O ⁸	
IP_DHCP_Server	OCTET STRING		O ⁸	

¹ Required to be read-only and shall contain the six octet B/IP address.

² Required to be present if, and only if, the port supports multicast.

³ Required to be present if, and only if, the port is capable of communicating through a NAT router as described in Clause J.7.5.

- ⁴ Required to be present if, and only if, the port is capable of functioning as a BBMD.
⁵ Required to be present if, and only if, the port is capable of functioning as a foreign device.
⁶ Shall be read-only if the IP_DHCP_Enable property is TRUE.
⁷ Required to be present if, and only if, the port can be configured by DHCP.
⁸ Shall be present only if the port can be configured by DHCP.

Table 12-71.4. Additional Properties of the Network Port Object Type if Network Type is IPV6

Property Identifier	Property Datatype	BACNET APPLICATION	PROTOCOL	PHYSICAL
Network_Number	Unsigned16	(R)		
Network_Number_Quality	BACnetNetworkNumberQuality	(R)		
APDU_Length	Unsigned	(R)		
MAC_Address	OCTET STRING	(R) ¹		
Virtual_MAC_Address_Table	BACnetLIST of BACnetVMACEntry	(R)		
BACnet_IPv6_Mode	BACnetIPMode	(R)		
BACnet_IPv6_UDP_Port	Unsigned16	(R)		
BACnet_IPv6_Multicast_Address	OCTET STRING	(R)		
Routing_Table	BACnetLIST of BACnetRouterEntry	O		
BBMD_Broadcast_Distribution_Table	BACnetLIST of BACnetBDTEntry	O ²		
BBMD_Accept_FD_Registrations	BOOLEAN	O ²		
BBMD_Foreign_Device_Table	BACnetLIST of BACnetFDTEEntry	O ²		
FD_BBMD_Address	BACnetHostNPort	O ³		
FD_Subscription_Lifetime	Unsigned16	O ³		
IPv6_Address	OCTET STRING		(R) ⁴	
IPv6_Prefix_Length	Unsigned8		(R) ⁴	
IPv6_Default_Gateway	OCTET STRING		(R) ⁴	
IPv6_DNS_Server	BACnetARRAY[N] of OCTET STRING		(R) ⁴	
IPv6_Auto_Addressing_Enable	BOOLEAN		O ⁵	
IPv6_DHCP_Lease_Time	Unsigned		O ⁶	
IPv6_DHCP_Lease_Time_Remaining	Unsigned		O ⁶	
IPv6_DHCP_Server	OCTET STRING		O ⁶	
IPv6_Zone_Index	CharacterString		O ⁷	

- ¹ Required to be read-only and shall contain the VMAC address.
² Required to be present if, and only if, the port is capable of functioning as a BBMD.
³ Required to be present if, and only if, the port is capable of functioning as a foreign device.
⁴ Required to be read-only if the value is configured by automatic address assignment.
⁵ Required to be present if, and only if, the port supports automatic IPv6 address assignment.
⁶ Shall be present only if the port supports auto-addressing with DHCPv6.
⁷ Required to be present if, and only if, the port supports multiple IPv6 link local addresses.

Table 12-71.5. Additional Properties of the Network Port Object Type when Network Type is LONTALK

Property Identifier	Property Datatype	BACNET APPLICATION	PROTOCOL	PHYSICAL
Network_Number	Unsigned16	(R)		
Network_Number_Quality	BACnetNetworkNumberQuality	(R)		
APDU_Length	Unsigned	(R)		
Routing_Table	BACnetLIST of BACnetRouterEntry	O		

<i>MAC_Address</i>	<i>OCTET STRING</i>			<i>(R)</i> ¹
<i>Link_Speed</i>	<i>REAL</i>			<i>O</i>
<i>Link_Speeds</i>	<i>BACnetARRAY[N] of REAL</i>			<i>O</i>
<i>Link_Speed_Autonegotiate</i>	<i>BOOLEAN</i>			<i>O</i>
<i>Network_Interface_Name</i>	<i>CharacterString</i>			<i>O</i>

¹ Required to be configurable.

Table 12-71.6. Additional Properties of the Network Port Object Type if Network Type is MSTP

<i>Property Identifier</i>	<i>Property Datatype</i>	<i>BACNET APPLICATION</i>	<i>PROTOCOL</i>	<i>PHYSICAL</i>
<i>Network_Number</i>	<i>Unsigned16</i>	<i>(R)</i>		
<i>Network_Number_Quality</i>	<i>BACnetNetworkNumberQuality</i>	<i>(R)</i>		
<i>APDU_Length</i>	<i>Unsigned</i>	<i>(R)</i>		
<i>Routing_Table</i>	<i>BACnetLIST of BACnetRouterEntry</i>	<i>O</i>		
<i>Slave_Proxy_Enable</i>	<i>BOOLEAN</i>	<i>O</i> ¹		
<i>Manual_Slave_Address_Binding</i>	<i>BACnetLIST of BACnetAddressBinding</i>	<i>O</i> ¹		
<i>Auto_Slave_Discovery</i>	<i>BOOLEAN</i>	<i>O</i> ²		
<i>Slave_Address_Binding</i>	<i>BACnetLIST of BACnetAddressBinding</i>	<i>O</i> ¹		
<i>MAC_Address</i>	<i>OCTET STRING</i>		<i>(R)</i> ³	
<i>Max_Master</i>	<i>Unsigned8</i>		<i>O</i> ⁴	
<i>Max_Info_Frames</i>	<i>Unsigned8</i>		<i>O</i> ⁴	

¹ Required to be present if, and only if, the port is capable of being a Slave-Proxy.

² Required to be present if, and only if, the port is capable of being a Slave-Proxy that implements automatic discovery of slaves.

³ Required to be configurable.

⁴ Required to be present if, and only if, the port is an MSTP master.

Table 12-71.7. Additional Properties of the Network Port Object Type if Network Type is PTP

<i>Property Identifier</i>	<i>Property Datatype</i>	<i>BACNET APPLICATION</i>	<i>PROTOCOL</i>	<i>PHYSICAL</i>
<i>APDU_Length</i>	<i>Unsigned</i>	<i>(R)</i>		
<i>Routing_Table</i>	<i>BACnetLIST of BACnetRouterEntry</i>	<i>O</i>		
<i>Link_Speed</i>	<i>REAL</i>			<i>(R)</i>
<i>Link_Speeds</i>	<i>BACnetARRAY[N] of REAL</i>			<i>O</i>
<i>Link_Speed_Autonegotiate</i>	<i>BOOLEAN</i>			<i>O</i>
<i>Network_Interface_Name</i>	<i>CharacterString</i>			<i>O</i>

[Note to reviewer: Network_Type SECURE_CONNECT is introduced with this addendum]

Table 12-71.8. Additional Properties of the Network Port Object Type if Network_Type is SECURE_CONNECT

<i>Property Identifier</i>	<i>Property Datatype</i>	<i>BACNET APPLICATION</i>	<i>PROTOCOL</i>	<i>PHYSICAL</i>
<i>Network_Number</i>	<i>Unsigned16</i>	(R)		
<i>Network_Number_Quality</i>	<i>BACnetNetworkNumberQuality</i>	(R)		
<i>APDU_Length</i>	<i>Unsigned</i>	(R)		
<i>MAC_Address</i>	<i>OCTET STRING</i>	(R) ¹		
<i>Max_BVLC_Length_Accepted</i>	<i>Unsigned</i>	(R)		
<i>Max_NPDU_Length_Accepted</i>	<i>Unsigned</i>	(R)		
<i>SC_Primary_Hub_URI</i>	<i>CharacterString</i>	(R)		
<i>SC_Failover_Hub_URI</i>	<i>CharacterString</i>	(R)		
<i>SC_Minimum_Reconnect_Time</i>	<i>Unsigned</i>	(R)		
<i>SC_Maximum_Reconnect_Time</i>	<i>Unsigned</i>	(R)		
<i>SC_Connect_Wait_Timeout</i>	<i>Unsigned</i>	(R)		
<i>SC_Disconnect_Wait_Timeout</i>	<i>Unsigned</i>	(R)		
<i>SC_Heartbeat_Timeout</i>	<i>Unsigned</i>	(R)		
<i>SC_Hub_Connector_State</i>	<i>BACnetSCHubConnectorState</i>	(R)		
<i>Operational_Certificate_File</i>	<i>BACnetObjectIdentifier</i>	(R)		
<i>Issuer_Certificate_Files</i>	<i>BACnetARRAY[2] of BACnetObjectIdentifier</i>	(R)		
<i>Certificate_Signing_Request_File</i>	<i>BACnetObjectIdentifier</i>	(R)		
<i>Routing_Table</i>	<i>BACnetLIST of BACnetRouterEntry</i>	O		
<i>SC_Primary_Hub_Connection_Status</i>	<i>BACnetSCHubConnection</i>	O		
<i>SC_Failover_Hub_Connection_Status</i>	<i>BACnetSCHubConnection</i>	O		
<i>SC_Hub_Function_Enable</i>	<i>BOOLEAN</i>	O ²		
<i>SC_Hub_Function_Accept_URIs</i>	<i>BACnetARRAY[N] of CharacterString</i>	O ²		
<i>SC_Hub_Function_Binding</i>	<i>CharacterString</i>	O ³		
<i>SC_Hub_Function_Connection_Status</i>	<i>BACnetLIST of BACnetSCHubFunctionConnection</i>	O ²		
<i>SC_Direct_Connect_Initiate_Enable</i>	<i>BOOLEAN</i>	O ⁴		
<i>SC_Direct_Connect_Accept_Enable</i>	<i>BOOLEAN</i>	O ⁵		
<i>SC_Direct_Connect_Accept_URIs</i>	<i>BACnetARRAY[N] of CharacterString</i>	O ⁵		
<i>SC_Direct_Connect_Binding</i>	<i>CharacterString</i>	O ⁶		
<i>SC_Direct_Connect_Connection_Status</i>	<i>BACnetLIST of BACnetSCDirectConnection</i>	O ⁷		
<i>SC_Failed_Connection_Requests</i>	<i>BACnetLIST of BACnetSCFailedConnectionRequest</i>	O ⁸		

¹ Shall contain the VMAC address.

² Required to be present if, and only if, the port supports the BACnet/SC hub function.

³ Shall be present only if the port supports the BACnet/SC hub function.

⁴ Required to be present if, and only if, the port supports initiating BACnet/SC direct connections.

⁵ Required to be present if, and only if, the port supports accepting BACnet/SC direct connections.

⁶ Shall be present only if the port supports accepting BACnet/SC direct connections.

⁷ Required to be present if, and only if, the port supports BACnet/SC direct connections.

⁸ Required to be present if, and only if, the port supports the BACnet/SC hub function or accepting BACnet/SC direct connections.

Table 12-71.9. Additional Properties of the Network Port Object Type if Network Type is SERIAL

Property Identifier	Property Datatype	BACNET_APPLICATION	PROTOCOL	PHYSICAL
<i>Link_Speed</i> <i>Link_Speeds</i> <i>Link_Speed_Autonegotiate</i> <i>Network_Interface_Name</i>	<i>REAL</i> <i>BACnetARRAY[N] of REAL</i> <i>BOOLEAN</i> <i>CharacterString</i>			(R) O O O

Table 12-71.10. Additional Properties of the Network Port Object Type if Network Type is VIRTUAL

Property Identifier	Property Datatype	BACNET_APPLICATION	PROTOCOL	PHYSICAL
<i>Network_Number</i> <i>Network_Number_Quality</i> <i>APDU_Length</i> <i>Routing_Table</i> <i>MAC_Address</i>	<i>Unsigned16</i> <i>BACnetNetworkNumberQuality</i> <i>Unsigned</i> <i>BACnetLIST of BACnetRouterEntry</i> <i>OCTET STRING</i>	(R) (R) (R) O O ¹		

¹ Required to be present in one of the Network Port objects in the hierarchy.

Table 12-71.11. Additional Properties of the Network Port Object Type if Network Type is ZIGBEE

Property Identifier	Property Datatype	BACNET_APPLICATION	PROTOCOL	PHYSICAL
<i>Network_Number</i> <i>Network_Number_Quality</i> <i>APDU_Length</i> <i>Virtual_MAC_Address_Table</i> <i>Routing_Table</i> <i>MAC_Address</i> <i>Link_Speed</i> <i>Link_Speeds</i> <i>Link_Speed_Autonegotiate</i> <i>Network_Interface_Name</i>	<i>Unsigned16</i> <i>BACnetNetworkNumberQuality</i> <i>Unsigned</i> <i>BACnetLIST of BACnetVMACEntry</i> <i>BACnetLIST of BACnetRouterEntry</i> <i>OCTET STRING</i> <i>REAL</i> <i>BACnetARRAY[N] of REAL</i> <i>BOOLEAN</i> <i>CharacterString</i>	(R) (R) (R) (R) O		(R) ¹ O O O O

¹ Required to be read-only and shall contain the VMAC address.

Table 12-71.12. Additional Properties of the Network Port Object Type if Network Type is proprietary

Property Identifier	Property Datatype	BACNET_APPLICATION	PROTOCOL	PHYSICAL
Network_Number	Unsigned16	(R)		
Network_Number_Quality	BACnetNetworkNumberQuality	(R)		
APDU_Length	Unsigned	(R)		
Routing_Table	BACnetLIST of BACnetRouterEntry	O		
MAC_Address	OCTET STRING	O ¹	O ¹	O ¹
<other standard properties>	<as defined for the property>	O ²	O ²	O ²

¹ Required to be present in one of the Network Port objects in the hierarchy if the topmost Network Port object has a Protocol_Level of BACNET_APPLICATION.

² Any standard property defined for this object type and not listed in this table may be present. If the presence or capability of one of these properties require other properties to be present, then the other properties shall be present.

Table 12-72. Required Properties of the Network Port Object Type Based on Network Type when Protocol_Level is BACNET_APPLICATION.

Network_Type	Additional Required Properties
ETHERNET	MAC_Address
IPV4 (BACnet_IP_Mode is NORMAL)	MAC_Address BACnet_IP_Mode IP_Address BACnet_IP_UDP_Port IP_Subnet_Mask IP_Default_Gateway IP_DNS_Server
IPV4 (BACnet_IP_Mode is FOREIGN)	MAC_Address BACnet_IP_Mode IP_Address BACnet_IP_UDP_Port IP_Subnet_Mask IP_Default_Gateway IP_DNS_Server FD_BBMD_Address FD_Subscription_Lifetime
IPV4 (BACnet_IP_Mode is BBMD)	MAC_Address BACnet_IP_Mode IP_Address BACnet_IP_UDP_Port IP_Subnet_Mask IP_Default_Gateway IP_DNS_Server BBMD_Broadcast_Distribution_Table BBMD_Accept_FD_Registrations BBMD_Foreign_Device_Table
IPV6 (BACnet_IPv6_Mode is NORMAL)	MAC_Address BACnet_IPv6_Mode

	IPv6_Prefix_Length IPv6_Address BACnet_IPv6_UDP_Port BACnet_IPv6_Multicast_Address IPv6_Default_Gateway IPv6_DNS_Server
IPv6 (BACnet_IPv6_Mode is FOREIGN)	MAC_Address BACnet_IPv6_Mode IPv6_Prefix_Length IPv6_Address BACnet_IPv6_UDP_Port BACnet_IPv6_Multicast_Address IPv6_Default_Gateway IPv6_DNS_Server FD_BBMD_Address FD_Subscription_Lifetime
IPv6 (BACnet_IPv6_Mode is BBMD)	MAC_Address BACnet_IPv6_Mode IPv6_Prefix_Length IPv6_Address BACnet_IPv6_UDP_Port BACnet_IPv6_Multicast_Address IPv6_Default_Gateway IPv6_DNS_Server BBMD_Broadcast_Distribution_Table BBMD_Accept_FD_Registrations BBMD_Foreign_Device_Table
MSTP (Slave node)	MAC_Address
MSTP (Master node)	MAC_Address
-	Max_Master
-	Max_Info_Frames
MSTP (capable of Slave Proxy)	MAC_Address
-	Max_Master
-	Max_Info_Frames
-	Slave_Proxy_Enable
-	Manual_Slave_Address_Binding
-	Auto_Slave_Discovery
-	Slave_Address_Binding

Table 12-73. Expected Properties of the Network Port Object Type by Network Type and Protocol Level.

Network Type	Protocol Level	Properties	Conformance
ARCNET	PHYSICAL	MAC_Address	R
ETHERNET		Link_Speed	R
LONTALK		Link_Speeds	O
VIRTUAL		Link_Speed_Autonegotiate	O
ZIGBEE		Network_Interface_Name	O
<proprietary values>			
SERIAL	PHYSICAL	Link_Speed	R
		Link_Speeds	O
		Link_Speed_Autonegotiate	O
		Network_Interface_Name	O
IPv4	PROTOCOL	IP_Address	R
		IP_Subnet_Mask	R

		IP_Default_Gateway IP_DNS_Server IP_DHCP_Enable IP_DHCP_Lease_Time IP_DHCP_Lease_Time_Remaining IP_DHCP_Server	R R O O O O
IPv6	PROTOCOL	IPv6_Address IPv6_Prefix_Length IPv6_Default_Gateway IPv6_DNS_Server IPv6_Auto_Addressing_Enable IPv6_DHCP_Lease_Time IPv6_DHCP_Lease_Time_Remaining IPv6_DHCP_Server IPv6_Zone_Index	R R R R O O O O O
MSTP (Slave node)	PROTOCOL	MAC_Address	R
MSTP (Master node)	PROTOCOL	MAC_Address Max_Master Max_Info_Frames	R R R
PTP	PROTOCOL		
<proprietary values>	PROTOCOL		
<proprietary values>	NON_BACNET_APPLICATION		
any (except SERIAL)	BACNET_APPLICATION	all properties	

Table 12-74. Properties of the Network Port Object Type Only Used when Protocol Level is BACNET_APPLICATION.

Network Type	Properties
ETHERNET ARCNET LONTALK PTP VIRTUAL <proprietary values>	
ZIGBEE	Virtual_MAC_Address_Table
MSTP	Slave_Proxy_Enable Manual_Slave_Address_Binding Auto_Slave_Discovery Slave_Address_Binding
IPv4	BACnet_IP_Mode BACnet_IP_UDP_Port BACnet_IP_Multicast_Address BACnet_IP_NAT_Traversal BACnet_IP_Global_Address BBMD_Broadcast_Distribution_Table BBMD_Accept_FD_Registrations BBMD_Foreign_Device_Table FD_BBMD_Address FD_Subscription_Lifetime
IPv6	BACnet_IPv6_Mode BACnet_IPv6_UDP_Port BACnet_IPv6_Multicast_Address BBMD_Broadcast_Distribution_Table BBMD_Accept_FD_Registrations BBMD_Foreign_Device_Table FD_BBMD_Address

	FD_Subscription_Lifetime
all	Network_Number Network_Number_Quality APDU_Length Routing_Table

[Change **Clause 12.56.8.** p. 548]

12.56.8 Network_Type

...

...	
VIRTUAL	Indicates that this port represents the configuration and properties of a virtual network as described in Clause H.2.
<i>SECURE_CONNECT</i>	<i>BACnet Secure Connect virtual link layer as defined in Annex AB.</i>
<Proprietary Enum Values>	A vendor may use other proprietary enumeration values to indicate that this port represents the use of message structures, procedures, and medium access control techniques other than those contained in this standard. For proprietary extensions of this enumeration, see Clause 23.1 of this standard.

...

[Change **Clause 12.56.9.** p. 549]

12.56.9 Protocol_Level

This property, of type BACnetProtocolLevel, indicates whether the object represents a physical network interface (PHYSICAL), a non-BACnet protocol (PROTOCOL), the BACnet use of the protocol (BACNET_APPLICATION), or a non-BACnet use of the protocol (NON_BACNET_APPLICATION).

If the Reference_Port property is absent, this property shall be BACNET_APPLICATION or NON_BACNET_APPLICATION.

[Change **Clause 12.56.10.** p. 549]

12.56.10 Reference_Port

This property, of type Unsigned, shall specify the instance of the Network Port object that this Network Port object uses as its lower protocol layer (i.e. transport, routing, data link, etc). This property allows the Network Port objects in the device to describe the hierarchy of protocols and physical ports in order to support complex network configuration required by some advanced BACnet products.

~~If this property is absent and the Protocol_Level is BACNET_APPLICATION, then this object is a non-hierarchical Network Port object that represents all protocol layers in a single object.~~

~~If this property has a value of 4194303 and the Protocol_Level is BACNET_APPLICATION, NON_BACNET_APPLICATION, or PROTOCOL, then this object has not been assigned a lower protocol layer. If the object is capable of representing all protocol layers in a single object, then this is a valid configuration and the object shall behave as if this property were absent. If the object is not capable of representing all protocol layers in a single object, then this is an indication that the object is not yet configured.~~

If Protocol_Level is PHYSICAL, this property shall be 4194303.

Object_Identifier	Network Port, 4
Object_Name	BACnetMSTP on USB1::COM1
Reference_Port	4194303
Protocol_Level	BACNET_APPLICATION
Network_Type	MSTP
Link_Speed	76800
Link_Speeds	9600,38400,76800
Link_Speed_Autonegotiate	FALSE
Network_Interface_Name	USB1::COM1
MAC_Address	1
Max_Master	12
Max_Info_Frames	3
Slave_Proxy_Enable	FALSE
Manual_Slave_Address_Binding	...
Auto_Slave_Discovery	FALSE
Slave_Address_Binding	...
Network_Number	40
Network_Number_Quality	CONFIGURED
APDU_Length	480
Routing_Table	...

Figure 12-18. Example Network Port With No Hierarchy Chain

A Network Port object is misconfigured if the referenced Network Port object has a Protocol_Level of BACNET_APPLICATION, or the referenced Network Port object does not exist.

If this property is writable, then a successful write to this property shall set the Changes_Pending property to TRUE. A value written to this property shall become effective when the device receives a ReinitializeDevice service request with a 'Reinitialized State of Device' of ACTIVATE_CHANGES or WARMSTART.

12.56.10.1 Network Port Hierarchies

Support for Network Port object hierarchies is optional.

In the normal case, a single hierarchy chain consists of a Network Port object with a Protocol_Level of PHYSICAL at the bottom; one or more Network Port objects with their Protocol_Level set to PROTOCOL, and a Network Port object with a Protocol_Level of BACNET_APPLICATION at the top.

In some cases, a higher level Network Port object may reference more than one lower level Network Port object. In that case, the Reference_Port property shall contain one of the referenced Network Port objects and the Additional_Reference_Ports property shall contain the remaining referenced Network Port objects.

Multiple Network Port objects can reference a PROTOCOL or PHYSICAL Network Port object.

A Network Port object with a Protocol_Level of BACNET_APPLICATION or PHYSICAL shall not be in the middle of a hierarchy chain.

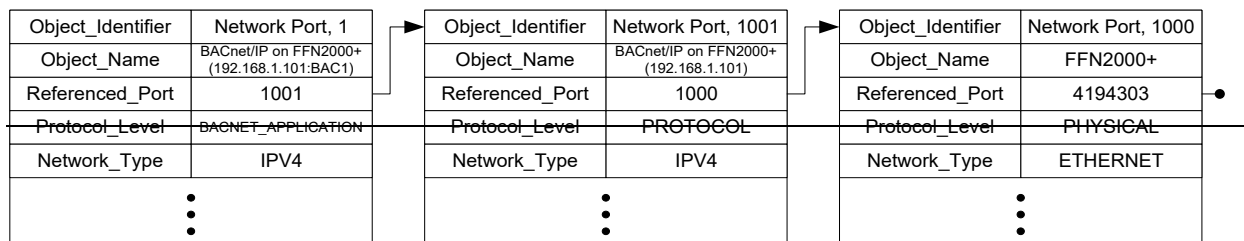


Figure 12-19. Example Network Port Hierarchy Chain

12.56.10.1.1 Property Inheritance

In a hierarchy chain of Network Port objects, a Network Port object with a Protocol_Level of BACNET_APPLICATION shall inherit property values related to configuration of the protocol or physical port from all Network Port objects in the chain.

Where a property is specified in multiple Network Port objects in the hierarchy chain, the property's value in the Network Port object nearest to the top of the chain shall be the value reflected in the topmost Network Port object. For example, in a Network Port object with a Protocol_Level of BACNET_APPLICATION, the Network_Type property shall be the same as the Network_Type of the directly referenced Network Port object.

Network Port objects with a Protocol_Level of NON_BACNET_APPLICATION are allowed to inherit property values. Network Port objects with a Protocol_Level other than BACNET_APPLICATION or NON_BACNET_APPLICATION shall not inherit property values.

Property inheritance allows clients to read and write current network settings by accessing the topmost Network Port object. It is required that inherited properties which are writable in the source (lower) Network Port object be writable in the inheriting Network Port object. When inherited properties are written, in either the source or the inheriting object, the values shall be written through to the other Network Port object. It is acceptable to make inherited properties in the source object read-only and the corresponding properties in the inheriting object be writable.

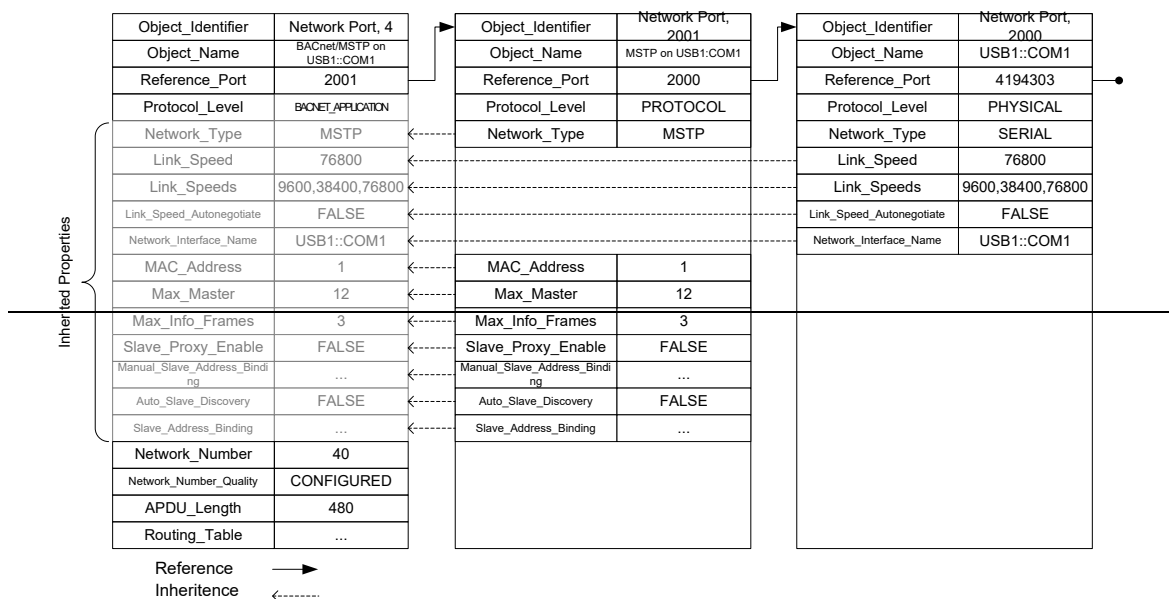


Figure 12-20. Example Network Port Hierarchy Chain Showing Property Inheritance

12.56.10.1.2 Pending Changes

In a hierarchy of Network Port objects, changes to a Network Port object may result in pending changes to multiple Network Port objects due to the sharing of inherited property values.

~~When an inherited property is written, and writes to the property are controlled via the pending changes functionality, it is a local matter whether or not the new property value is reflected in all affected objects immediately, or only once the changes have been activated. If the property value shows up immediately in all affected objects, then the Changes_Pending property in each of those objects shall be set to TRUE.~~

[Change Clause 12.56.11. p. 551]

12.56.11 Network_Number

...

~~If the Network_Type is PTP, then this property shall be read-only and contain a value of 0.~~

...

[Change Clause 12.56.12. p. 551]

12.56.12 Network_Number_Quality

This read-only property, of type BACnetNetworkNumberQuality, represents the current quality of the Network_Number property. ~~If the Network_Type is PTP, the Network_Number_Quality shall be CONFIGURED.~~

...

[Change Clause 12.56.14. p. 552]

12.56.14 Command

...

If the value of the Changes_Pending property is TRUE, writing a *supported command* value other than DISCARD_CHANGES or VALIDATE_CHANGES ~~(if supported)~~ to the Command property shall result in the return of a Result(-) with an 'Error Class' of PROPERTY and an 'Error Code' of INVALID_VALUE_IN_THIS_STATE.

...

...

GENERATE_CSR_FILE

If this port supports this command, this command shall generate a new certificate signing request file. The new certificate signing request file shall be referenced from the Certificate_Signing_Request_File property. See Clause 12.56.Y26. The new certificate signing request file shall contain a new public key from a newly generated public/private key pair.

The Command shall remain unchanged until the certificate signing request file has been generated at which point the Command shall be set to IDLE. This Command shall not affect the Changes_Pending property.

If this port does not support this command, writing this value shall result in the return of a Result(-) with an 'Error Class' of PROPERTY and an 'Error Code' of OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED.

VALIDATE_CHANGES

If this port supports this command, this command shall initiate a validation of the values of the properties of this port as specified in each property. If a property is present but not used, based on the Network_Type, it shall not be validated. The value of the Command_Validation_Result property shall be updated to indicate the validation result. See Clause 12.56.Y28.

The Command shall remain unchanged until the validation has been completed at which point the Command shall be set to IDLE. This Command shall not affect the Changes_Pending property.

If the device supports pending changes then the port shall support this command.

If the device does not support this command, writing this command shall return Result(-) with an 'Error Class' of PROPERTY and an 'Error Code' of OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED.

...

[Change Clause 12.56.15. p. 555]

12.56.15 MAC_Address

This property, of type OCTET STRING, contains the BACnet MAC address used on this network. The value of this property shall be conveyed with the most significant octet first. ~~If Network_Type is IPV4 and the Protocol_Level is BACNET_APPLICATION, then the value of this property shall contain the six octet combination of the IP_Address and BACnet_IP_UDP_Port and shall be read-only. If the value of Network_Type is a value that represents a port that requires VMAC addressing, then the value of this property shall be read-only and contain the VMAC address.~~

When a port is represented by a single non-hierarchical Network Port object, this property is required except when Network_Type is PTP.

...

[Change Clause 12.56.18. p. 556]

12.56.18 Link_Speeds

This read-only property, of type BACnetARRAY of REAL, is an array of the link speeds supported by this network port.

This property shall be present, if and only if, Link_Speed is present and writable.

[Change Clause 12.56.20. p. 556]

12.56.20 Network_Interface_Name

This property, of type CharacterString, is used to identify the network interface ~~hardware~~ to which this network port is bound. For example, if Network_Type is ETHERNET, the value of this property identifies the *name for the network interface* (e.g., "eth0", "Ethernet 2", "enp2s0") ~~Ethernet hardware interface~~ that this network port is using to communicate.

~~This property shall be read-only if it is inherited from another Network Port object.~~

If this property is writable, then a successful write to this property shall set the Changes_Pending property to TRUE. A value written to this property shall become effective when the device receives a ReinitializeDevice service request with a 'Reinitialized State of Device' of ACTIVATE_CHANGES or WARMSTART.

[Add new Properties to the Network Port object for BACnet/SC]

12.56.Y1 Additional_Reference_Ports

This property, of type BACnetLIST of Unsigned, contains a list of one or more Network Port object instances that this Network Port object uses as its lower protocol layer. This property extends the Reference_Port property by allowing references to additional Network Port objects. See Clause 12.56.

If the Reference_Port property is absent this property shall be absent, and this Network Port object shall represent all protocol layers in a single object.

If Protocol_Level is BACNET_APPLICATION, NON_BACNET_APPLICATION or PROTOCOL, an empty list indicates this property has not been assigned a lower protocol layer.

If Protocol_Level is PHYSICAL, this property shall be an empty list.

A Network Port object is misconfigured if an entry contains 4194303, a referenced Network Port object has a Protocol_Level of BACNET_APPLICATION, or the referenced Network Port object does not exist.

If this property is writable, then a successful write to this property shall set the Changes_Pending property to TRUE. A value written to this property shall become effective when the device receives a ReinitializeDevice service request with a 'Reinitialized State of Device' of ACTIVATE_CHANGES or WARMSTART.

12.56.Y2 Max_BVLC_Length_Accepted

This property, of type Unsigned, indicates the maximum size in octets of the BVLC message the network port can receive and process. A value of zero means that the maximum size is unknown or fixed by the Network_Type.

12.56.Y3 Max_NPDU_Length_Accepted

This property, of type Unsigned, indicates the maximum size in octets of the NPDU message the network port can receive and process. See Table 6-1.

12.56.Y4 SC_Primary_Hub_URI

This property, of type CharacterString, provides the URI for the primary hub function. The value shall be a UTF-8 string containing a WebSocket URI as of RFC 6455, formatted as of RFC 3986. If no URI is configured, this property shall contain an empty UTF-8 string. See Clauses AB.5.2 and AB.5.4. This property shall be configurable.

If this property is writable, then a successful write to this property shall set the Changes_Pending property to TRUE. A value written to this property shall become effective when the device receives a ReinitializeDevice service request with a 'Reinitialized State of Device' of ACTIVATE_CHANGES or WARMSTART.

12.56.Y5 SC_Failover_Hub_URI

This property, of type CharacterString, provides the URI for the failover hub function. The value shall be a UTF-8 string containing a WebSocket URI as of RFC 6455, formatted as of RFC 3986. If no URI is configured, this property shall contain an empty UTF-8 string. See Clauses AB.5.2 and AB.5.4. This property shall be configurable.

If this property is writable, then a successful write to this property shall set the Changes_Pending property to TRUE. A value written to this property shall become effective when the device receives a ReinitializeDevice service request with a 'Reinitialized State of Device' of ACTIVATE_CHANGES or WARMSTART.

12.56.Y6 SC_Minimum_Reconnect_Time

This property, of type Unsigned, specifies the minimum time in seconds that the initiating peer shall wait between attempts to establish a BACnet/SC connection.

Reconnection strategies and delay between attempts to connect to other BACnet/SC nodes are generally a local matter. See Clause AB.6.1.

This property shall be configurable and support a range of 2..300 seconds.

If the value of the SC_Minimum_Reconnect_Time is greater than the value of the SC_Maximum_Reconnect_Time, it is a local matter how long to wait between each connection attempt, but the time shall not exceed the value of the SC_Maximum_Reconnect_Time.

12.56.Y7 SC_Maximum_Reconnect_Time

This property, of type Unsigned, specifies the maximum time in seconds that the initiating peer shall wait between attempts to establish a BACnet/SC connection.

Reconnection strategies and delay between attempts to connect to other BACnet/SC nodes are generally a local matter. See Clause AB.6.1.

This property shall be configurable and support a range of 2..600 seconds.

12.56.Y8 SC_Connect_Wait_Timeout

This property, of type Unsigned, identifies the time in seconds that a node waits for a response BVLC message while connecting a BACnet/SC connection. See Clause AB.6.2.

This property shall be configurable and support a minimum range of 5..300 seconds. The recommended default value is 10 seconds.

12.56.Y9 SC_Disconnect_Wait_Timeout

This property, of type Unsigned, identifies the time in seconds that a node waits for a response BVLC message while disconnecting from a BACnet/SC connection. See Clause AB.6.2.

This property shall be configurable and support a minimum range of 5..300 seconds. The recommended default value is 10 seconds.

12.56.Y10 SC_Heartbeat_Timeout

This property, of type Unsigned, identifies the time in seconds of received BVLC message inactivity on a BACnet/SC connection before a Heartbeat-Request message is sent by an initiating BACnet/SC node. See Clause AB.6.3.

This property shall be configurable and support a range of 3..300 seconds. The recommended default value is 300 seconds.

12.56.Y11 SC_Hub_Connector_State

This read-only property, of type BACnetSCHubConnectorState, indicates the state of the BACnet/SC node's hub connector. The following states are defined:

NO_HUB_CONNECTION	A connection has not been established
CONNECTED_TO_PRIMARY	Connected to the primary hub
CONNECTED_TO_FAILOVER	Connected to the failover hub

12.56.Y12 SC_Primary_Hub_Connection_Status

This read-only property, of type BACnetSCHubConnection, indicates the status and associated information about the primary hub connection initiated by the hub connector. The BACnetSCHubConnection structure has the following fields:

Connection State	This field, of type BACnetSCConnectionState, indicates the current state of the hub connection. NOT_CONNECTED – The connection was successfully disconnected, or no connection has been attempted. CONNECTED – The connection state is CONNECTED. See Clause AB.6.2.2. DISCONNECTED_WITH_ERRORS – The connection disconnected with errors. FAILED_TO_CONNECT – The connection attempt failed. See Clause AB.6.2.2.
Connect Timestamp	This field, of type BACnetDateTime, indicates the local date and time when the connection was established, or the connection attempt failed. If the connection has never been established, this field shall contain an unspecified datetime.
Disconnect Timestamp	This field, of type BACnetDateTime, indicates the local date and time when the connection successfully disconnected or disconnected with errors. If the connection has never disconnected, this field shall contain an unspecified datetime.
Error	This field, of type Error, indicates the most recent error for the connection. If the 'Connection State' field is NOT_CONNECTED or CONNECTED, this field shall be absent.
Error Details	This optional field, of type CharacterString, provides details on the most recent error. If the 'Connection State' field is NOT_CONNECTED or CONNECTED, this field shall be absent.

12.56.Y13 SC_Failover_Hub_Connection_Status

This read-only property, of type BACnetSCHubConnection, indicates the status and associated information about the failover hub connection initiated by the hub connector. The BACnetSCHubConnection structure is described in Clause 12.56.Y12.

12.56.Y14 SC_Hub_Function_Enable

This property, of type BOOLEAN, indicates and determines whether (TRUE) or not (FALSE) this port is performing the BACnet/SC hub function. This property shall be configurable.

If this property is writable, then a successful write to this property shall set the Changes_Pending property to TRUE. A value written to this property shall become effective when the device receives a ReinitializeDevice service request with a 'Reinitialized State of Device' of ACTIVATE_CHANGES or WARMSTART.

12.56.Y15 SC_Hub_Function_Accept_URIs

This property, of type BACnetARRAY of CharacterString, contains an array of WebSocket URIs that represents the possible URIs a BACnet/SC hub connector could use to connect to the hub function. This property provides a BACnet visible way to store and present the possible BACnet/SC hub function WebSocket URIs. This property does not affect the configuration of the port nor is it generated by the port.

Each WebSocket URI entry shall be a UTF-8 string containing a WebSocket URI as of RFC 6455, formatted as of RFC 3986. This property shall be configurable.

12.56.Y16 SC_Hub_Function_Binding

This property, of type `CharacterString`, is used to configure the detailed binding information for the network port. The content and syntax of this string is a local matter and is product or application specific. For example, in the simplest case, this could just be a port number, “4443”, but other networks ports might require more details, e.g., “eth0:4443,eth1:8080”.

If this property is writable, then a successful write to this property shall set the `Changes_Pending` property to `TRUE`. A value written to this property shall become effective when the device receives a `ReinitializeDevice` service request with a 'Reinitialized State of Device' of `ACTIVATE_CHANGES` or `WARMSTART`.

12.56.Y17 SC_Hub_Function_Connection_Status

This read-only property, of type `BACnetLIST` of `BACnetSCHubFunctionConnection`, indicates the status and associated information about all BACnet/SC connections accepted by the hub function. It is a local matter how long an entry will remain after the connection is no longer active. The `BACnetSCHubFunctionConnection` structure has the following fields:

Connection State	This field, of type <code>BACnetSCConnectionState</code>, indicates the current state of the hub connection. NOT_CONNECTED – The connection was successfully disconnected. CONNECTED – The connection state is <code>CONNECTED</code>. See Clause AB.6.2.3. DISCONNECTED_WITH_ERRORS – The connection disconnected with errors.
Connect Timestamp	This field, of type <code>BACnetDateTime</code> , indicates the local date and time when the connection was established.
Disconnect Timestamp	This field, of type <code>BACnetDateTime</code>, indicates the local date and time when the connection successfully disconnected or disconnected with errors. If the connection has never disconnected, this field shall contain an unspecified datetime.
Peer Address	This field, of type <code>BACnetHostNPort</code> , indicates the address of the peer node.
Peer VMAC	This field, of type <code>OCTET STRING (SIZE(6))</code> , is the VMAC address of the peer node.
Peer UUID	This field, of type <code>OCTET STRING (SIZE(16))</code> , is the device UUID of the peer node.
Error	This field, of type <code>Error</code> , indicates the most recent error for the connection. If the ‘Connection State’ field is <code>NOT_CONNECTED</code> or <code>CONNECTED</code> , this field shall be absent.
Error Details	This optional field, of type <code>CharacterString</code> , provides details on the most recent error. If the ‘Connection State’ field is <code>NOT_CONNECTED</code> or <code>CONNECTED</code> , this field shall be absent.

12.56.Y18 SC_Direct_Connect_Initiate_Enable

This property, of type `BOOLEAN`, indicates and controls whether (`TRUE`) or not (`FALSE`) this port is enabled to initiate BACnet/SC Direct Connections. This property shall be writable.

A successful write to this property shall set the `Changes_Pending` property to `TRUE`. A value written to this property shall become effective when the device receives a `ReinitializeDevice` service request with a 'Reinitialized State of Device' of `ACTIVATE_CHANGES` or `WARMSTART`.

12.56.Y19 SC_Direct_Connect_Accept_Enable

This property, of type `BOOLEAN`, indicates and controls whether (`TRUE`) or not (`FALSE`) this port is enabled to accept BACnet/SC Direct Connections. This property shall be writable.

A successful write to this property shall set the `Changes_Pending` property to `TRUE`. A value written to this property shall become effective when the device receives a `ReinitializeDevice` service request with a 'Reinitialized State of Device' of `ACTIVATE_CHANGES` or `WARMSTART`.

12.56.Y20 SC_Direct_Connect_Accept_URIs

This property, of type `BACnetARRAY` of `CharacterString` contains an array of WebSocket URIs that are included in the Address-Resolution-Ack response. This array represents the possible URIs an initiating BACnet/SC direct connection device could use to directly connect to this device. This property does not affect the configuration of the port nor is it generated by the port.

Each WebSocket URI entry shall be a UTF-8 string containing a WebSocket URI as of RFC 6455, formatted as of RFC 3986. This property shall be configurable.

12.56.Y21 SC_Direct_Connect_Binding

This property, of type `CharacterString`, is used to configure the detailed binding information for the network port. The content and syntax of this string is a local matter and is product or application specific. For example, in the simplest case, this could just be a port number, "4443", but other networks ports might require more details, e.g., "eth0:4443,eth1:8080".

If this property is writable, then a successful write to this property shall set the `Changes_Pending` property to `TRUE`. A value written to this property shall become effective when the device receives a `ReinitializeDevice` service request with a 'Reinitialized State of Device' of `ACTIVATE_CHANGES` or `WARMSTART`.

12.56.Y22 SC_Direct_Connect_Connection_Status

This read-only property, of type `BACnetLIST` of `BACnetSCDirectConnection`, indicates the status and associated information about BACnet/SC direct connections initiated and accepted by this port.

When the port is initiating a direct connection, an entry shall be added to this property. It is a local matter how long the entry will remain after the connection is no longer active.

When the port has accepted a direct connection, an entry shall be added to this property. It is a local matter how long the entry remains after disconnection.

The `BACnetSCDirectConnection` structure has the following fields:

URI	This field, of type <code>CharacterString</code> , indicates the WebSocket URI for the direct connection.
	For initiated direct connections, this field shall contain the WebSocket URI that was used for initiating the connection.
	For accepted direct connections, this field shall be an empty string.

Connection State	This field, of type BACnetSCConnectionState, indicates the current state of the direct connection. NOT_CONNECTED – The connection was successfully disconnected, or no connection has been attempted. CONNECTED – The connection state is CONNECTED. See Clause AB.6.2.2 for initiated direct connections and See Clause AB.6.2.3 for accepted direct connections. DISCONNECTED_WITH_ERRORS – The connection disconnected with errors. FAILED_TO_CONNECT – For initiated direct connections, this indicates the connection attempt failed. See Clause AB.6.2.2. Not valid for accepting connections.
Connect Timestamp	This field, of type BACnetDateTime, indicates the local date and time when the connection was established, or the connection attempt failed. If the connection has never been established, this field shall contain an unspecified datetime.
Disconnect Timestamp	This field, of type BACnetDateTime, indicates the local date and time when the connection successfully disconnected or disconnected with errors. If the connection has never disconnected, this field shall contain an unspecified datetime.
Peer Address	This field, of type BACnetHostNPort, indicates the address of the peer node. If the peer node's Address has not been determined, this field shall be absent.
Peer VMAC	This field, of type OCTET STRING (SIZE(6)), is the VMAC address of the peer node. If the peer node's VMAC has not been determined, this field shall be absent.
Peer UUID	This field, of type OCTET STRING (SIZE(16)), is the device UUID of the peer node. If the peer node's Device UUID has not been determined, this field shall be absent.
Error	This field, of type Error, indicates the most recent error for the connection. If the 'Connection State' field is NOT_CONNECTED or CONNECTED, this field shall be absent.
Error Details	This optional field, of type CharacterString, provides details on the most recent error. If the 'Connection State' field is NOT_CONNECTED or CONNECTED, this field shall be absent.

12.56.Y23 SC_Failed_Connection_Requests

This read-only property, of type BACnetLIST of BACnetSCFailedConnectionRequest, indicates information about failed connection requests received by an enabled hub function or a node switch enabled for accepting a BACnet/SC direct connection. This property shall support at least one entry.

Due to variations in implementations at the lower levels of the protocol stack, only errors resulting from the exchange of BVLC messages are required to be included in this property. Inclusion of other detected errors is a local matter.

If a new failed connection request occurs that matches the IP Address of the Peer Address parameter of an existing entry, that entry shall be overwritten with the information from the new failed connection request.

If a new failed connection request occurs that does not match the IP Address of the Peer Address parameter of any of the existing entries and the list cannot support another entry, the oldest entry shall be overwritten with the information from the new failed connection request. It is a local matter how long an entry remains in the list.

The BACnetSCFailedConnectionRequest structure has the following fields:

Timestamp	This field, of type BACnetDateTime, indicates the local date and time of the connection request.
Peer Address	This field, of type BACnetHostNPort, indicates the address of the peer node.
Peer VMAC	This field, of type OCTET STRING (SIZE(6)), is the VMAC address of the peer node. If the peer node's VMAC has not been determined, this field shall be absent.
Peer UUID	This field, of type OCTET STRING (SIZE(16)), is the device UUID of the peer node. If the peer node's Device UUID has not been determined, this field shall be absent.
Error	This field, of type Error, indicates the most recent error for the connection.
Error Details	This optional field, of type CharacterString, provides details on the most recent error.

12.56.Y24 Operational_Certificate_File

This read-only property, of type BACnetObjectIdentifier, specifies the File object that contains the operational certificate used to identify the IP host of the port. The operational certificate file shall be an X.509 certificate in PEM format. For details on a BACnet/SC network see Clause AB.7.4.1.

The minimum file size supported for the operational certificate shall be large enough to store the largest encryption formats plus a minimum of 1000 octets of additional storage for optional certificate parameters (such as subject alternate names, etc).

The file data in the File object referenced by this property shall be writable. If the referenced File object does not contain an operational certificate, the value of the File_Size property of the File object shall be 0.

Writing VALIDATE_CHANGES to the Command property shall, at a minimum, perform the below required validations on this property. If any of the validations fail, the Property field in the Command_Validation_Result property shall contain Operational_Certificate_File and the Error field shall contain an 'Error Class' and 'Error Code' as specified below.

<u>Required Standard Validations</u>	<u>Error Class</u>	<u>Error Code</u>
The operational certificate cannot be consumed because of an encoding error.	SECURITY	CERTIFICATE_MALFORMED
The operational certificate file failed to validate against an internal private key.	SECURITY	UNKNOWN_KEY
The operational certificate cannot be validated by one of the issuer certificates.	SECURITY	CERTIFICATE_INVALID
The operational certificate file has expired.	SECURITY	CERTIFICATE_EXPIRED

Optional Standard Validations

The operational certificate file has been revoked.

Error Class

SECURITY

Error Code

CERTIFICATE_REVOKED

A successful change to the file data in the File object referenced by this property shall set the `Changes_Pending` property to TRUE and the new file data shall become effective when the device receives a `ReinitializeDevice` service request with a 'Reinitialized State of Device' of `ACTIVATE_CHANGES` or `WARMSTART`. If the `DISCARD_CHANGES` command is supported by the `Command` property for this port, writing a value of `DISCARD_CHANGES` shall revert the file data in the File object referenced by this property to the file data when `Changes_Pending` was last equal to FALSE.

An attempt to write to this property shall result in a `Result(-)` to be returned with an 'error class' of `PROPERTY` and an 'error code' of `WRITE_ACCESS_DENIED`.

12.56.Y25 Issuer_Certificate_Files

This read-only property, of type `BACnetARRAY[2]` of `BACnetObjectIdentifier`, specifies the File objects that contain the signing CA issuer certificate files the port accepts when it is establishing a secure connection. Each issuer certificate file shall contain an X.509 certificate in PEM format. For details on a BACnet/SC network see Clause AB.7.4.1.1.

This property contains two array elements and each element shall reference a File object within the device. The file data in the File object shall be writable. If a referenced File object does not contain an issuer certificate, the value of the `File_Size` property of the File object shall be 0.

The device shall be capable of storing two issuer certificate files for this port. The minimum file size supported for each issuer certificate shall be large enough to store the largest encryption format plus a minimum of 1000 octets of additional storage for optional certificate parameters (such as subject alternate names, etc.).

A successful change in the file data of a File object referenced by this property will set the `Changes_Pending` property to TRUE and the new file data shall become effective when the device receives a `ReinitializeDevice` service request with a 'Reinitialized State of Device' of `ACTIVATE_CHANGES` or `WARMSTART`. If the `DISCARD_CHANGES` command is supported by the `Command` property for this port, writing a value of `DISCARD_CHANGES` shall revert the file data in the File objects referenced by this property to the file data when `Changes_Pending` was last equal to FALSE.

An attempt to write the entire property, the size of the array, or the value of an element of this property shall result in a `Result(-)` to be returned with an 'error class' of `PROPERTY` and an 'error code' of `WRITE_ACCESS_DENIED`.

12.56.Y26 Certificate_Signing_Request_File

This read-only property, of type `BACnetObjectIdentifier`, specifies the File object that contains the PKCS #10 certificate signing request as defined by RFC 5967 in PEM format that can be used by the signing CA to create an operational certificate, matching the internal private key for the port. For details on a BACnet/SC network see Clause AB.7.4.1.2.

If the `GENERATE_CSR_FILE` command is supported, the certificate signing request file shall be updated and a new private/public key pair shall be generated when the `Command` property is written to `GENERATE_CSR_FILE`. The certificate signing request file shall contain the X.509 'subject' distinguished name specified in the currently active operational certificate, see Clause 12.56.Y24, and the public key from the new private/public key pair. If no operational certificate is currently active, then the X.509 'subject' distinguished name of the certificate signing request is a local matter but shall be globally unique.

The certificate signing request file is read by an external entity and is provided to a signing CA to create an operational certificate. The selection of the signing CA is dictated by site policy. The resulting operational certificate is then written by the external entity into the port's operational certificate file.

If the referenced File object does not contain a valid certificate signing request file, the value of the `File_Size` property of the File object shall be 0.

If the `GENERATE_CSR_FILE` command is not supported, then the certificate signing request file shall be present and the X.509 'subject' distinguished name of the certificate signing request is a local matter but shall be globally unique.

12.56.Y27 Current_Health

This read-only property, of type BACnetPortHealth, indicates the current health of this port. It is a local matter when and how often this port is evaluated. When multiple error conditions exist, it is a local matter which is reported in this property.

This property shall not report errors caused by pending changes.

The BACnetHealth structure has the following fields:

Timestamp	This field, of type BACnetDateTime, indicates the local date and time when this property was last updated. If the device has no knowledge of local time or date, this field shall contain an unspecified datetime.
Result	This field, of type Error, indicates the most recent error for this port. If the port is not currently in error, this field shall contain 'Error Class' of OBJECT and 'Error Code' of SUCCESS. If this port is not in error but a referenced port is in error, this field shall contain 'Error Class' of OBJECT and 'Error Code' of REFERENCED_PORT_IN_ERROR.
Property	This optional field, of type BACnetPropertyIdentifier, indicates the property that caused the error. If this port is not currently in error, this field shall be absent.
Details	This optional field, of type CharacterString, provides details about the most recent error. If the port is not currently in error, this field shall be absent.

12.56.Y28 Command_Validation_Result

This read-only property, of type BACnetHealth, indicates any errors detected as a result of writing VALIDATE_CHANGES to the Command property. This property shall be present if the VALIDATE_CHANGES command is supported. See Clause 12.56.13. The BACnetHealth structure is described in Clause 12.56.Y27.

[Add new error codes to **Clause 18.2**, p. 790]

...

REFERENCED_PORT_IN_ERROR – *The referenced Network Port object is in error.*

...

[Add new error codes to **Clause 18.5**, p. 793]

...

CERTIFICATE_EXPIRED – *The operational certificate file has expired or is not yet valid.*

CERTIFICATE_MALFORMED – *The operational certificate file contains an encoding error.*

CERTIFICATE_REVOKED – *The operational certificate file has been revoked.*

CERTIFICATE_INVALID – *The operational certificate file cannot be validated by any configured issuer certificates.*

UNKNOWN_KEY – *The operational certificate file does not match a private key.*

...

[Clause 21]

[Add new productions to **Clause 21.6** section "**Base Types**" maintaining the alphabetical order, p. 882]

```
BACnetHealth ::= SEQUENCE {
    timestamp                [0] BACnetDateTime,
    result                    [1] Error,
    property                  [2] BACnetPropertyIdentifier OPTIONAL,
    details                   [3] CharacterString OPTIONAL
}
```

```
BACnetSCConnectionState ::= ENUMERATED {
    not-connected            (0),
    connected                (1),
    disconnected-with-errors  (2),
    failed-to-connect        (3)
}
```

```
BACnetSCDirectConnection ::= SEQUENCE {
    uri                      [0] CharacterString,
    connection-state         [1] BACnetSCConnectionState,
    connect-timestamp        [2] BACnetDateTime,
    disconnect-timestamp     [3] BACnetDateTime,
    peer-address             [4] BACnetHostNPort OPTIONAL,
    peer-vmac                [5] OCTET STRING (SIZE(6)) OPTIONAL,
    peer-uuid                [6] OCTET STRING (SIZE(16)) OPTIONAL,
    error                    [7] Error OPTIONAL,
    error-details            [8] CharacterString OPTIONAL
}
```

```
BACnetSCFailedConnectionRequest ::= SEQUENCE {
    timestamp                [0] BACnetDateTime,
    peer-address             [1] BACnetHostNPort,
    peer-vmac                [2] OCTET STRING (SIZE(6)) OPTIONAL,
    peer-uuid                [3] OCTET STRING (SIZE(16)) OPTIONAL,
    error                    [4] Error,
    error-details            [5] CharacterString OPTIONAL
}
```

```
BACnetSCHubConnection ::= SEQUENCE {
    connection-state         [0] BACnetSCConnectionState,
    connect-timestamp        [1] BACnetDateTime,
    disconnect-timestamp     [2] BACnetDateTime,
    error                    [3] Error OPTIONAL,
    error-details            [4] CharacterString OPTIONAL
}
```

```
BACnetSCHubConnectorState ::= ENUMERATED {
    no-hub-connection        (0),
    connected-to-primary      (1),
    connected-to-failover     (2),
}
```

```
BACnetSCHubFunctionConnection ::= SEQUENCE {
    connection-state         [0] BACnetSCConnectionState,
    connect-timestamp        [1] BACnetDateTime,
    disconnect-timestamp     [2] BACnetDateTime,
    peer-address             [3] BACnetHostNPort,
```

```

peer-vmac          [4] OCTET STRING (SIZE(6)),
peer-uuid          [5] OCTET STRING (SIZE(16)),
error              [6] Error OPTIONAL,
error-details      [7] CharacterString OPTIONAL
}

```

BACnetNetworkPortCommand ::= ENUMERATED {

```

...
restart-port      (7),
generate-csr-file (8),
validate-changes  (9),
...
}

```

-- Enumerated values 0-127 are reserved for definition by ASHRAE. Enumerated values
-- 128-255 may be used by others subject to the procedures and constraints described
-- in Clause 23.

BACnetNetworkType ::= ENUMERATED {

```

...
serial           (10),
secure-connect    (11),
...
}

```

-- Enumerated values 0-63 are reserved for definition by ASHRAE. Enumerated values
-- 64-255 may be used by others subject to the procedures and constraints described
-- in Clause 23.

[Add new error codes to the **Error** production in **Clause 21**, p. 798]

Error ::= SEQUENCE {

-- NOTE: The valid combinations of error-class and error-code are defined in Clause 18.

error-class ENUMERATED {

```

...
},

```

-- Enumerated values 0-63 are reserved for definition by ASHRAE. Enumerated values
-- 64-65535 may be used by others subject to the procedures and constraints described
-- in Clause 23.

error-code ENUMERATED { -- see below for numerical order

```

...

```

[Insert here the new error codes as added in the numerical order list below into the alphabetical order list, maintaining the alphabetical order]

```

...

```

```

character-set-not-supported      (41),
certificate-expired              (200),
certificate-invalid              (201),
certificate-malformed            (202),
certificate-revoked              (203),
...

```

```

unknown-file-size               (122),
unknown-key                     (204),
...

```

-- numerical order reference

```

...

```

```

-- see certificate-expired      (200),

```

-- *see certificate-invalid* (201),
 -- *see certificate-malformed* (202),
 -- *see certificate-revoked* (203),
 -- *see unknown-key* (204),
 ...

[Insert into production **BACnetPropertyIdentifier** in **Clause 21.6**, preserving the alphabetical and numerical order, p. 925]

...
 actual-shed-level (212),
 additional-reference-ports (508),
 adjust-value (176),
 ...
 car-position (458),
 certificate-signing-request-file (509),
 ...
 client-cov-increment (127),
 command-validation-result (510),
 command (417),
 ...
 current-command-priority (431),
 current-health (4194307),
 database-revision (155),
 ...
 issue-confirmed-notifications (51),
 issuer-certificate-files (511),
 is-utc (344),
 ...
 max-apdu-length-accepted (62),
 max-bvlc-length-accepted (4194304),
 max-failed-attempts (285),
 ...
 max-master (64),
 max-mpdu-length-accepted (4194305),
 max-pres-value (65),
 ...
 occupancy-upper-limit-enforced (298),
 operational-certificate-file (4194306),
 operation-direction (477),
 ...
 routing-table (428),
 sc-connect-wait-timeout (4194308),
 sc-direct-connect-accept-enable (4194309),
 sc-direct-connect-accept-uris (4194310),
 sc-direct-connect-binding (4194311),
 sc-direct-connect-connection-status (4194312),
 sc-direct-connect-initiate-enable (4194313),
 sc-disconnect-wait-timeout (4194314),
 sc-failed-connection-requests (4194315),
 sc-failover-hub-connection-status (4194316),
 sc-failover-hub-uri (4194317),
 sc-hub-connector-state (4194318),
 sc-hub-function-accept-uris (4194319),
 sc-hub-function-binding (4194320),
 sc-hub-function-connection-status (4194321),
 sc-hub-function-enable (4194322),
 sc-heartbeat-timeout (4194323),

```

    sc-primary-hub-connection-status      (4194324),
    sc-primary-hub-uri                    (4194325),
    sc-maximum-reconnect-time             (4194326),
    sc-minimum-reconnect-time             (4194327),
    scale                                 (187),
    ...
-- numerical order reference
    ...
    -- see additional-reference-ports      (508),
    -- see certificate-signing-request-file (509),
    -- see command-validation-result       (510),
    -- see issuer-certificate-files        (511),
    -- see max-bvlc-length-accepted        (4194304),
    -- see max-npdu-length-accepted        (4194305),
    -- see operational-certificate-file     (4194306),
    -- see current-health                  (4194307),
    -- see sc-connect-wait-timeout          (4194308),
    -- see sc-direct-connect-accept-enable (4194309),
    -- see sc-direct-connect-accept-uris   (4194310),
    -- see sc-direct-connect-binding        (4194311),
    -- see sc-direct-connect-connection-status (4194312),
    -- see sc-direct-connect-initiate-enable (4194313),
    -- see sc-disconnect-wait-timeout       (4194314),
    -- see sc-failed-connection-requests   (4194315),
    -- see sc-failover-hub-connection-status (4194316),
    -- see sc-failover-hub-uri             (4194317),
    -- see sc-hub-connector-state           (4194318),
    -- see sc-hub-function-accept-uris     (4194319),
    -- see sc-hub-function-binding          (4194320),
    -- see sc-hub-function-connection-status (4194321),
    -- see sc-hub-function-enable           (4194322),
    -- see sc-heartbeat-timeout             (4194323),
    -- see sc-primary-hub-connection-status (4194324),
    -- see sc-primary-hub-uri              (4194325),
    -- see sc-maximum-reconnect-time       (4194326),
    -- see sc-minimum-reconnect-time       (4194327),
    ...
}
-- The special property identifiers ...

```

[Insert into production **BACnetPropertyStates** in **Clause 21.6**, p. 942]

```

    sc-connection-state      [258] BACnetSCConnectionState,
    sc-hub-connector-state    [259] BACnetSCHubConnectorState,

```

135-2020cc-2. Modifications to Annex AB

Rationale

Annex AB needs changes to better match the proposed changes to the Network Port object to support BACnet/SC.

[Change Clause AB.1.8. p. 1383]

[Note to reviewer, the 'x' in the following refers to the value of the Protocol_Revision of this addendum when published]

AB.1.8 BACnet/SC Network Port Objects

...

For BACnet/SC network port implementations with a ~~protocol revision~~ *Protocol_Revision* 17 and higher through 23, BACnet/SC network ports shall be represented by a Network Port object at the BACNET_APPLICATION protocol level with a proprietary network type value. For the standard properties that are required to be present, see Clause 12.56.

For BACnet/SC network port implementations with a Protocol_Revision 24 and higher, BACnet/SC network ports shall be represented by a Network Port object at the BACNET_APPLICATION protocol level with network type of SECURE_CONNECT. For the required standard properties to be present see Clause 12.56.

[Change Clause AB.5.3.1. p. 1399]

AB.5.3.1 Hub Connections

...

All BVLC message types related to the hub connection shall be initiated and sent to connection peers. See Clause AB.6.2. The VMAC address provided by the hub in a Connect-Accept message shall be the VMAC address of the BACnet/SC node of the network port in which the hub function resides. Note that the VMAC is provided to the hub connector to allow a client to know the device which is hosting the hub function. It is otherwise unused by this protocol.

If the hub function determines that the connection or the connection peer might not be alive, the hub function shall test with a Heartbeat-Request prior to terminating the connection. See Clause AB.2.14.

The hub function's URI on which it accepts hub connections is a WebSocket URI identified by the "wss" scheme.

...

[Change Clause AB.6.1. p. 1401]

AB.6.1 BACnet/SC Reconnect Timeout

...

Increasing reconnect timeouts should be applied between unsuccessful attempts to connect. The algorithm for increasing is a local matter, however the reconnect timeout shall not be increased beyond 600 seconds.

As of Protocol_Revision 24, the minimum reconnect timeout shall be configurable.

[Change Clause AB.6.3. p. 1405]

AB.6.3 Connection Keep-Alive

...

On receipt of Heartbeat-Request, the accepting peer shall respond with a Heartbeat-ACK message to the initiating peer.

As of Protocol_Revision 24, the heartbeat timeout shall be configurable and if a Heartbeat-ACK message is not received from the accepting peer, the initiating peer shall initiate the 'Local disconnection' procedure. See Clause AB.6.2.2.

If the heartbeat timeout is configurable, it shall support a minimum range of 3..300 seconds. A fixed heartbeat timeout shall have a value in the range 30..300 seconds.

...

[Change Clause AB.7.4. p. 1406]

AB.7.4.1 Certificate Management

Secure WebSocket connections require the use of TLS. The creation of private keys, public certificates and the management of the certificate signing authority, or authorities, are site-specific deployment options beyond the scope of this standard. However, to ensure interoperability, a BACnet/SC implementation shall support the storage and use of certificates as defined in the following ~~subclauses~~ clauses.

Devices claiming Protocol_Revision 24 or greater shall support certificate configuration through the Network Port object.

AB.7.4.1.3 Configuring Operational Certificates

...

A device that supports an internal security function that allows it to generate and store its private keys by itself is not allowed to expose the private keys, and may not be allowed to accept a private key from a configuration tool. To create a signed operational certificate, the configuration tool provides certificate parameters of the installation to the device and initiates a private key and certificate signing request generation by the device. The certificate signing request is sent to a signing of the installation. The signed certificate returned from the CA, and the CA certificates for the accepted CAs as-required for the installation are configured into the device by the tool.

If the effective operational certificate of an active connection is changed, the connection shall be re-established.

If an issuer certificate is removed from the set of effective issuer certificates and the issuer certificate was used, or might have been used, to validate a peer's certificate for a connection, the connection shall be re-established.

135-2020cc-3. Add a Procedure to Replace BACnet/SC Certificates

Rationale

BACnet/SC certificates will usually have a finite lifetime. A procedure to replace expiring certificates is required to ensure the network is minimally interrupted during this transition.

[add to **Clause 3** in alphabetical order]

CSR certificate signing request (see RFC 2986)

[add new **Clause 19.Y**, p. 827]

19.Y Replacing Certificates in a BACnet/SC Network

This clause describes the procedures used to replace certificates in a BACnet/SC network.

These procedures assume that the BACnet/SC devices to be configured contain a Network Port object with Network_Type of SECURE_CONNECT, the BACnet/SC network is active, each node has a valid operational certificate, all operational certificates validate against the same issuer certificate, and each node has that issuer certificate in its set of active issuer certificates.

During procedures that update issuer certificates, a second issuer certificate will be distributed to all nodes before operational certificates are updated so that nodes are able to validate both initial and new operational certificates.

19.Y.1 Conditions and Overview of Certificate Changes

This clause provides several examples of when certificates would require updating.

19.Y.1.1 Expiration of an Operational Certificate

Operational certificates can be valid for a limited period of time. Assuming that all operational certificates have the same, or similar, expiration dates, when the operational certificates are close to the expiration date, the operational certificate in every node in a BACnet/SC network will need to be replaced with a new operational certificate that has an updated expiration date. See the procedure in clause 19.Y.2.1.

19.Y.1.2 Expiration of an Issuer Certificate

Issuer certificates can be valid for a limited period of time. When the issuer certificate is close to the expiration date, both the issuer certificate and the operational certificate need to be replaced in every node in a BACnet/SC network. See the procedure in clause 19.Y.2.2.

19.Y.1.3 Excluding a Node from the Network

If a node must be removed from a BACnet/SC network but cannot be physically disconnected, follow the procedure in clause 19.Y.2.3.

19.Y.2 Updating Procedures

19.Y.2.1 Extending Operational Certificates' Validity Period

Before the operational certificates expires:

1. Replace the operational certificate in every node in the network as described in Clause 19.Y.3.2.

19.Y.2.2 Extending Issuer Certificate Validity Period

Before the issuer certificate expires:

1. Add a new issuer certificate to every node in the network as described in 19.Y.3.1.
2. Replace the operational certificate in every node in the network as described in 19.Y.3.2.
3. Optionally, remove the old issuer certificate from every node in the network as described in 19.Y.3.3.

Strictly following the sequence is important. Failure to do so may result in some nodes becoming unable to participate in the BACnet/SC network.

When performing the above steps, do not advance to the next step until the previous step has successfully completed. Performing steps before previous steps complete will result in nodes which cannot be updated through the BACnet/SC network thus requiring access via an alternate data link or physical access to the node. See Clause 19.Y.3.4.

19.Y.2.3 Banning One or More Nodes from the Network

When one or more nodes need to be removed from a network but cannot be physically disconnected from the network, all other nodes have to be moved to a new issuer certificate. To achieve this, follow these steps:

1. Add a new issuer certificate to each node in the network as described in 19.Y.3.1 skipping the nodes to be removed.
2. Replace the operational certificates in all nodes as described in 19.Y.3.2 skipping the nodes to be removed.
3. Remove the old issuer certificate from each node in the network as described in 19.Y.3.3 skipping the removed nodes.

Strictly following the sequence is important. Failure to do so may result in some nodes becoming unable to participate in the BACnet/SC network.

When performing the above steps, do not advance to the next step until the previous step has successfully completed. Performing steps before previous steps complete will result in nodes which cannot be updated through the BACnet/SC network thus requiring access via an alternate data link or physical access to the node.

19.Y.3 Procedural Steps Description

In the following procedural descriptions, the device performing the procedures in this clause will be referred to as device A.

19.Y.3.1 Adding a New Issuer Certificate to the Network

This procedure is used to add a new issuer certificate to the network and is used in conjunction with other procedures in order to ensure that the network remains accessible. As such, it is important to not remove or replace the currently used issuer certificate, otherwise the nodes will no longer be able to connect to the network due to certification validation failures when the issuer certificate that signed the operational certificate is missing.

In BACnet/SC Network Port objects there are two issuer certificate files. This procedure assumes that the network is already up and running and as such one of those files contains the currently active issuer certificate. The other issuer certificate file will either be empty or contain an outdated issuer certificate. The procedure will replace the content of this empty or outdated issuer certificate file.

To add the new issuer certificate:

1. Device A determines the File object that represents the currently used issuer certificate for this network by reading the file content of one of the issuer certificate files and comparing it to the known current issuer certificate.
2. Device A writes the new issuer certificate to the certificate file that represents the other issuer certificate.
3. Device A sends a ReinitializeDevice(ACTIVATE_CHANGES or WARMSTART, <password>) message to activate the changes to the Network Port object.

These steps need to be applied to all nodes in the network, including hubs and temporarily unavailable nodes. Any nodes which cannot be updated will be unable to participate in the BACnet/SC network once operational certificates are updated. Such nodes will need to be updated by some other secure method. See Clause 19.Y.3.4.

The order in which the nodes are updated is not important.

19.Y.3.2 Replace the Operational Certificate

There are two procedures for replacing operational certificates. Which procedure is used depends on whether the node is capable of generating a certificate signing request file, or whether the node contains a fixed certificate signing request file.

For nodes capable of generating an internal private/public key pair:

1. If a new private/public key pair is required, device A writes the GENERATE_CSR_FILE command to the Command property.
2. Device A uploads the File object that contains the CSR from the node.
3. Device A requests the signing CA generate a new operational certificate based on the CSR data from step 2.
4. Device A writes the File object that represents the operational certificate for this node.
5. Device A sends a ReinitializeDevice(ACTIVATE_CHANGES or WARMSTART, <password>) message to activate the changes to the Network Port object.

For nodes that contain a fixed CSR:

1. Device A uploads the File object that contains the CSR from the node.
2. Device A requests the signing CA generate a new operational certificate based on the CSR data from step 1.
3. Device A writes the File object that represents the operational certificate for this node.
4. Device A sends a ReinitializeDevice(ACTIVATE_CHANGES or WARMSTART, <password>) message to activate the changes to the Network Port object.

These steps need to be applied to all nodes in the network, including hubs and temporarily unavailable nodes. Any nodes which cannot be updated will be unable to participate in the BACnet/SC network once operational certificates are updated. Such nodes will need to be updated by some other secure method. See Clause 19.Y.3.4.

This procedure may be applied to the nodes in any order and may be applied to the nodes in parallel.

19.Y.3.3 Removing an Outdated Issuer Certificate from the Network

In order to not adversely affect the network, it is important to not remove the currently used issuer certificate, otherwise the nodes will no longer be able to connect to the network.

1. Device A determines the File object that represents the currently used issuer certificate for this network by reading the file content of one of the issuer certificate files and comparing it to the known current issuer certificate.
2. Device A writes zero to the File_Size property of the issuer certificate File object that represents the other issuer certificate.
3. Device A sends a ReinitializeDevice(ACTIVATE_CHANGES or WARMSTART, <password>) message to activate the changes to the Network Port object.

These steps need to be applied to all nodes in the network, including hubs and temporarily unavailable nodes. Any nodes which cannot be updated will be unable to participate in the BACnet/SC network once operational certificates are updated. Such nodes will need to be updated by some other secure method. See Clause 19.Y.3.4.

The order in which the nodes are addressed is not important.

19.Y.3.4 Repairing Nodes Inadvertently Removed from the Network

It is expected that there will be times when some nodes are offline and the certificate update process cannot be delayed. In these cases, the offline nodes will need to be updated later when they are available. There are multiple ways in which an offline node can be brought back online:

1. The node supports incoming direct connections, in which case the certificate management tool can connect via direct connect using an outdated operational certificate and matching issuer certificate.
2. The node is a router and can be configured via one of its other secure datalinks.
3. The node is physically removed, connected to an alternate physical network on which the certificate management tool will play the role of DHCP server, DNS server, and BACnet/SC hub at the expected URI.
4. The node is physically accessed, reset to factory defaults as defined in Clause AB.7.4.2, and re-configured using its most recent backup.

5. The node provides a secure proprietary method of configuration. This method shall be documented and can be disabled by the customer.

Option 3 should work except where the node's operational certificate or issuer certificate has expired. Given that physical access to the device may be difficult and / or expensive, every reasonable effort should be made to ensure that all devices are able to be updated when issuer certificates and / or operational certificates are updated.

135-2020cc-4. Add Network Port Object Configuration BIBBs

Rationale

There is currently no mechanism to specify the ability to configure a Network Port object.

There is currently no mechanism to specify the ability to interoperably replace BACnet/SC certificates.

[Add **Clauses K.6.Y1** and **K.6.Y2**. p. 1115]

K.6.Y1 BIBB – Network Management-Communications Configuration-A (NM-CC-A)

The A device is able to present and configure the properties of the Network Port object of device B.

Devices claiming conformance to NM-CC-A shall support initiating ReinitializeDevice requests containing the Password parameter and are required to support the ACTIVATE_CHANGES and WARMSTART service choices.

The A device shall be capable of using ReadProperty to retrieve all standard properties of the Network Port object type except any property defined by the standard as not readable via ReadProperty. Device A may use alternate services where support for execution of the alternate service is supported by Device B.

The A Device shall be capable of using WriteProperty to modify all standard properties of the Network port object type except any property defined by the standard as read-only, or to which access is otherwise restricted by the standard.

The A device shall use AtomicReadFile and AtomicWriteFile services to retrieve and modify the file data in all File objects referenced by standard properties in the Network Port object.

BACnet Service	Initiate	Execute
WriteProperty	x	
ReadProperty	x	
AtomicWriteFile	x	
AtomicReadFile	x	
ReinitializeDevice	x	

K.6.Y2 BIBB – Network Management-Secure Connect Certificate Management-A (NM-SCCM-A)

The A device manages the certificates of a BACnet/SC network using all of the procedures described in 19.Y.

BACnet Service	Initiate	Execute
WriteProperty	x	
ReadProperty	x	
AtomicWriteFile	x	
AtomicReadFile	x	
ReinitializeDevice	x	

[Change **Clause L.1**. p. 1115]

L.1 Operator Interface Profiles

The following table indicates which BIBBs shall be supported by the device types of this family, for each interoperability area. The B-XAWS is not shown in this table and is described in Clause L.1.1.

...

Device & Network Management

B-AWS	B-OWS	B-OD
DM-DDB-A,B	DM-DDB-A,B	DM-DDB-A,B
DM-ANM-A		
DM-ADM-A		
DM-DOB-B	DM-DOB-B	DM-DOB-B
DM-DCC-A		
DM-MTS-A	DM-MTS-A	
DM-OCD-A		
DM-RD-A		
DM-BR-A		
DM-DDA-A		
<i>NM-CC-A</i>		

Audit Reporting

B-AWS	B-OWS	B-OD
AR-AVM-A		

[Add a new entry to **History of Revisions**, p. 1429]

(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	24	Addendum cc to ANSI/ASHRAE 135-2020 Approved by the ASHRAE Standards Committee MONTH X, 20XX; by the ASHRAE Board of Directors MONTH X, 20XX; and by the American National Standards Institute MONTH X, 20XX. 1. Update the Network Port Object and Add BACnet/SC Configuration Support. 2. Modifications to Annex AB. 3. Add a Procedure to Replace BACnet/SC Certificates. 4. Add Network Port Object Configuration BIBBs.

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.

ASHRAE · 180 Technology Parkway NW · Peachtree Corners, GA 30092 · www.ashrae.org

About ASHRAE

Founded in 1894, ASHRAE is a global professional society committed to serve humanity by advancing the arts and sciences of heating, ventilation, air conditioning, refrigeration, and their allied fields.

As an industry leader in research, standards writing, publishing, certification, and continuing education, ASHRAE and its members are dedicated to promoting a healthy and sustainable built environment for all, through strategic partnerships with organizations in the HVAC&R community and across related industries.

To stay current with this and other ASHRAE Standards and Guidelines, visit www.ashrae.org/standards, and connect on LinkedIn, Facebook, Twitter, and YouTube.

Visit the ASHRAE Bookstore

ASHRAE offers its Standards and Guidelines in print, as immediately downloadable PDFs, and via ASHRAE Digital Collections, which provides online access with automatic updates as well as historical versions of publications. Selected Standards and Guidelines are also offered in redline versions that indicate the changes made between the active Standard or Guideline and its previous version. For more information, visit the Standards and Guidelines section of the ASHRAE Bookstore at www.ashrae.org/bookstore.

IMPORTANT NOTICES ABOUT THIS STANDARD

To ensure that you have all of the approved addenda, errata, and interpretations for this Standard, visit www.ashrae.org/standards to download them free of charge.

Addenda, errata, and interpretations for ASHRAE Standards and Guidelines are no longer distributed with copies of the Standards and Guidelines. ASHRAE provides these addenda, errata, and interpretations only in electronic form to promote more sustainable use of resources.