



**BSR/ASHRAE Addendum cc to
ANSI/ASHRAE Standard 135-2016**

Public Review Draft

Proposed Addendum cc to Standard 135-2016, BACnet[®] - A Data Communication Protocol for Building Automation and Control Networks

**First Public Review (April 2020)
(Draft shows Proposed Changes to Current Standard)**

This draft has been recommended for public review by the responsible project committee. To submit a comment on this proposed standard, go to the ASHRAE website at www.ashrae.org/standards-research--technology/public-review-drafts and access the online comment database. The draft is subject to modification until it is approved for publication by the Board of Directors and ANSI. Until this time, the current edition of the standard (as modified by any published addenda on the ASHRAE website) remains in effect. The current edition of any standard may be purchased from the ASHRAE Online Store at www.ashrae.org/bookstore or by calling 404-636-8400 or 1-800-727-4723 (for orders in the U.S. or Canada).

This standard is under continuous maintenance. To propose a change to the current standard, use the change submittal form available on the ASHRAE website, www.ashrae.org.

The appearance of any technical data or editorial material in this public review document does not constitute endorsement, warranty, or guaranty by ASHRAE of any product, service, process, procedure, or design, and ASHRAE expressly disclaims such.

© 2020 ASHRAE. This draft is covered under ASHRAE copyright. Permission to reproduce or redistribute all or any part of this document must be obtained from the ASHRAE Manager of Standards, 1791 Tullie Circle, NE, Atlanta, GA 30329. Phone: 404-636-8400, Ext. 1125. Fax: 404-321-5478. E-mail: standards.section@ashrae.org.

ASHRAE, 1791 Tullie Circle, NE, Atlanta GA 30329-2305

[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.

135-2016*cc*-1. Extend the Network Port Object Type for BACnet/SC, p. 3.

In the following document, language to be added to existing clauses of ANSI/ASHRAE 135-2016 and Addenda is indicated through the use of *italics*, while deletions are indicated by ~~striketrough~~. 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.

135-2016cc-1. Add BACnet/SC Configuration Support

Rationale

135-2016 Addendum *bj* (BACnet/SC) requires some configuration items, status indications, and other elements to be network visible and interoperability 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. 517]

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, i/t 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.~~

Table 12-71. 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
Network_Number	Unsigned16	R ¹
Network_Number_Quality	BACnetNetworkNumberQuality	R
Changes_Pending	BOOLEAN	R
Command	BACnetNetworkPortCommand	O ²
MAC_Address	OCTET STRING	O ³
APDU_Length	Unsigned	R
Link_Speed	REAL	R
Link_Speeds	BACnetARRAY[N] of REAL	O ⁴
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	O
IP_DHCP_Lease_Time_Remaining	Unsigned	O
IP_DHCP_Server	OCTET STRING	O
BACnet_IP_NAT_Traversal	BOOLEAN	O ⁹
BACnet_IP_Global_Address	BACnetHostNPort	O ¹⁰
BBMD_Broadcast_Distribution_Table	BACnetLIST of BACnetBDTEntry	O ¹¹
BBMD_Accept_FD_Registrations	BOOLEAN	O ¹¹
BBMD_Foreign_Device_Table	BACnetLIST of BACnetFDTEntry	O ¹²
FD_BBMD_Address	BACnetHostNPort	O ¹³
FD_Subscription_Lifetime	Unsigned16	O ¹³
BACnet_IPv6_Mode	BACnetIPMode	O ¹⁴
IPv6_Address	OCTET STRING	O ¹⁵
IPv6_Prefix_Length	Unsigned8	O ¹⁵
BACnet_IPv6_UDP_Port	Unsigned16	O ¹⁴
IPv6_Default_Gateway	OCTET STRING	O ¹⁵
BACnet_IPv6_Multicast_Address	OCTET STRING	O ¹⁴
IPv6_DNS_Server	BACnetARRAY[N] of OCTET STRING	O ¹⁵
IPv6_Auto_Addressing_Enable	BOOLEAN	O ¹⁶
IPv6_DHCP_Lease_Time	Unsigned	O
IPv6_DHCP_Lease_Time_Remaining	Unsigned	O
IPv6_DHCP_Server	OCTET STRING	O
IPv6_Zone_Index	CharacterString	O ¹⁷
Max_Master	Unsigned8(0..127)	O ¹⁸
Max_Info_Frames	Unsigned8	O ¹⁸
Slave_Proxy_Enable	BOOLEAN	O ¹⁹
Manual_Slave_Address_Binding	BACnetLIST of BACnetAddressBinding	O ¹⁹
Auto_Slave_Discovery	BOOLEAN	O ²⁰
Slave_Address_Binding	BACnetLIST of BACnetAddressBinding	O ²¹
Virtual_MAC_Address_Table	BACnetLIST of BACnetVMACEntry	O ²²
Max_BVLC_Length_Accepted	Unsigned	O ²³
Max_NPDU_Length_Accepted	Unsigned	O ²³
SC_Primary_Hub_URI	CharacterString	O ²³
SC_Failover_Hub_URI	CharacterString	O ²³
SC_Minimum_Reconnect_Time	Unsigned	O ²³
SC_Maximum_Reconnect_Time	Unsigned	O ²³

<i>SC_Connection_Wait_Timeout</i>	<i>Unsigned</i>	<i>O²³</i>
<i>SC_Disconnect_Wait_Timeout</i>	<i>Unsigned</i>	<i>O²³</i>
<i>SC_Initiating_Heartbeat_Timeout</i>	<i>Unsigned</i>	<i>O²³</i>
<i>SC_Accepting_Heartbeat_Timeout</i>	<i>Unsigned</i>	<i>O²³</i>
<i>SC_Hub_Connector_State</i>	<i>BACnetSCHubConnectionState</i>	<i>O²³</i>
<i>SC_Primary_Hub_Status</i>	<i>BACnetSCHubConnection</i>	<i>O</i>
<i>SC_Failover_Hub_Status</i>	<i>BACnetSCHubConnection</i>	<i>O</i>
<i>SC_Hub_Function_Enable</i>	<i>BOOLEAN</i>	<i>O^{24,25}</i>
<i>SC_Hub_Function_Accept_URIs</i>	<i>BACnetARRAY[N] of CharacterString</i>	<i>O^{24,25}</i>
<i>SC_Hub_Function_Connection_Status</i>	<i>BACnetARRAY[N] of BACnetSCHubFunctionConnection</i>	<i>O²⁴</i>
<i>SC_Direct_Connect_Accept_URIs</i>	<i>BACnetARRAY[N] of CharacterString</i>	<i>O^{26,27}</i>
<i>SC_Direct_Connect_Connection_Status</i>	<i>BACnetARRAY[N] of BACnetSCDirectConnect</i>	<i>O²⁶</i>
<i>SC_Reference_Ports</i>	<i>BACnetLIST of Unsigned</i>	<i>O</i>
<i>Operational_Certificate</i>	<i>OCTET STRING</i>	<i>O^{23,28}</i>
<i>CA_Certificates</i>	<i>BACnetARRAY[N] of OCTET STRING</i>	<i>O^{23,28}</i>
<i>Private_Key</i>	<i>OCTET STRING</i>	<i>O^{29,31}</i>
<i>Certificate_Signing_Parameters</i>	<i>BACnetCertificateSigningParameters</i>	<i>O³⁰</i>
<i>Certificate_Signing_Request</i>	<i>OCTET STRING</i>	<i>O^{30,31}</i>
<i>Routing_Table</i>	<i>BACnetLIST of BACnetRouterEntry</i>	<i>O</i>
<i>Event_Detection_Enable</i>	<i>BOOLEAN</i>	<i>O^{23,24,32,33}</i>
<i>Notification_Class</i>	<i>Unsigned</i>	<i>O^{32,33,23,24}</i>
<i>Event_Enable</i>	<i>BACnetEventTransitionBits</i>	<i>O^{32,33,23,24}</i>
<i>Acked_Transitions</i>	<i>BACnetEventTransitionBits</i>	<i>O^{32,33,23,24}</i>
<i>Notify_Type</i>	<i>BACnetNotifyType</i>	<i>O^{32,33,23,24}</i>
<i>Event_Time_Stamps</i>	<i>BACnetARRAY[3] of BACnetTimeStamp</i>	<i>O^{32,33,23,24}</i>
<i>Event_Message_Texts</i>	<i>BACnetARRAY[3] of CharacterString</i>	<i>O^{33,24}</i>
<i>Event_Message_Texts_Config</i>	<i>BACnetARRAY[3] of CharacterString</i>	<i>O^{33,24}</i>
<i>Event_State</i>	<i>BACnetEventState</i>	<i>O^{32,23}</i>
<i>Reliability_Evaluation_Inhibit</i>	<i>BOOLEAN</i>	<i>O</i>
<i>Property_List</i>	<i>BACnetARRAY[N] of BACnetPropertyIdentifier</i>	<i>R</i>
<i>Tags</i>	<i>BACnetARRAY[N] of BACnetNameValue</i>	<i>O</i>
<i>Profile_Location</i>	<i>CharacterString</i>	<i>O</i>
<i>Profile_Name</i>	<i>CharacterString</i>	<i>O</i>

- ¹ Required to be writable in routers, secure devices, and any other device that requires knowledge of the network number for proper operation.
- ² Shall 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 device 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.
- 23 *Required if the Network Type is SECURE_CONNECT.*
- 24 *Required to be present if the port supports the BACnet/SC hub function.*
- 25 *Required to be configurable if the port supports the BACnet/SC hub function.*
- 26 *Required to be present if the port supports BACnet/SC direct connections.*
- 27 *Required to be configurable if the port supports BACnet/SC direct connections.*
- 28 *Required to be configurable.*
- 29 *If the Network Type is SECURE_CONNECT, this property is required if the Certificate_Signing_Request property is absent.*
- 30 *If the Network Type is SECURE_CONNECT, this property is required if the Private_Key property is absent.*
- 31 *If present, this property shall be configurable.*
- 322 These properties are required if the object supports intrinsic reporting.
- 3
- 332 These properties shall be present only if the object supports intrinsic reporting.
- 4

[Network Port object Property Changes]

[Change Clause 12.56.8. p. 523]

12.56.8 Network_Type

...

ARCNET	
IPV4	
IPV6	
ETHERNET	
LONTALK	
MSTP	MS/TP, as defined in Clause 9.
PTP	Point-To-Point, as defined in Clause 10.
SERIAL	A physical serial port.
ZIGBEE	
VIRTUAL	Indicates that this port represents the configuration and properties of a virtual network as described in Clause H.2.
SECURE_CONNECT	<i>BACnet Secure Connect virtual link layer as defined in 135-2016 Addendum bj.</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. 524]

12.56.9 Protocol_Level

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

[Change **Clause 12.56.10**. p. 524]

12.56.10 Reference_Port

...

If this property has a value of 4194303, 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, and *Protocol_Level* is not *PHYSICAL*, then this is an indication that the object is not yet configured.

If Protocol_Level has a value of PHYSICAL, then this property shall have a value of 4194303.

If Network_Type is SECURE_CONNECT, then this property shall have a value of 4194303. Since BACnet Secure Connect is URI based, multiple IP network ports could be used. See Clause 12.56.Y16.

...

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

...

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 or NON_BACNET_APPLICATION at the top. Multiple Network Port objects can reference a PROTOCOL or PHYSICAL Network Port object.

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

...

[Add new Clauses 12.56 for Network Port Properties for BACnet/SC]

12.56.Y1 Max_BVLC_Length_Accepted

This property, of type Unsigned, shall indicate 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.Y2 Max_NPDU_Length_Accepted

This property, of type Unsigned, shall indicate the maximum size in octets of the NPDU message the network port can receive and process.

12.56.Y3 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.

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.Y4 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.

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_Minimum_Reconnect_Time

This property, of type Unsigned, specifies the minimum time in seconds between each attempt to establish a BACnet/SC connection.

This property shall be configurable.

Reconnection strategies and delay between attempts to connect to other BACnet/SC nodes are generally a local matter. The minimum time between attempts to establish a connection shall be as specified by this property.

If this property is writable, it shall support a minimum range of 2..300 seconds.

12.56.Y6 SC_Maximum_Reconnect_Time

This property, of type Unsigned, specifies the maximum time in seconds between each attempt to establish a BACnet/SC connection.

This property shall be configurable.

Reconnection strategies and delay between attempts to connect to other BACnet/SC nodes are generally a local matter. The maximum time between attempts to establish a connection shall be as specified by this property.

If this property is writable, it shall support a minimum range of 2..300 seconds.

12.56.Y7 SC_Connection_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.

This property shall be configurable.

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

12.56.Y8 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.

This property shall be configurable.

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

12.56.Y9 SC_Initiating_Heartbeat_Timeout

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

This property shall be configurable.

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

12.56.Y10 SC_Accepting_Heartbeat_Timeout

This property, of type Unsigned, identifies the time in seconds of BVLC message inactivity on a BACnet/SC connection before an accepting BACnet/SC node will drop the connection. Note that proper operation requires that this be configured to be sufficiently greater than the SC_Initiating_Heartbeat_Timeout of all the nodes that initiate connections to this node.

This property shall be configurable.

This property shall support a minimum range of 3..500. The recommended default value is 500 seconds.

12.56.Y11 SC_Hub_Connector_State

This property, of type BACnetSCHubConnectorState, shall indicate if the BACnet/SC node's hub connector is connected and, if so, to the primary or failover hub. This property shall be:

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_Status

This property, of type BACnetSCHubConnection, shall indicate 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, shall indicate the current state of the hub connection. Idle – The last connection was successfully disconnected. Connected – The connection is active. Failed - The last connection disconnected with errors.
Error	This optional field, of type Error, shall indicate the most recent error for the connection. This field shall not represent any error reported through a BVLC-Result NAK message received through the connection. If the 'Connection State' field is Idle or Connected, this field shall be absent.
Details	This optional field, of type CharacterString, may be used to indicate details on the most recent error. If the 'Connection State' field is Idle or Connected, this field shall be absent.

12.56.Y13 SC_Failover_Hub_Status

This property, of type BACnetSCHubConnection, shall indicate the status and associated information about the failover hub connection initiated by the hub connector. The BACnetSCHubConnection structure has the following fields:

Connection State	This field, of type BACnetSCConnectionState, shall indicate the current state of the hub connection. Idle – The last connection was successfully disconnected. Connected – The connection is active. Failed - The last connection disconnected with errors.
Error	This optional field, of type Error, shall indicate the most recent error for the connection. This field shall not represent any error reported through a BVLC-Result NAK message received through the connection. If the 'Connection State' field is Idle or Connected, this field shall be absent.
Details	This optional field, of type CharacterString, may be used to indicate details on the most recent error. If the 'Connection State' field is Idle or Connected, this field shall be absent.

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.

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[N] of CharacterString contains an array of WebSocket URIs that represents the possible URIs a BACnet/SC hub connector device 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. 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 if the port supports the BACnet/SC hub function.

12.56.Y16 SC_Hub_Function_Connection_Status

This property, of type BACnetARRAY[N] of BACnetSCHubFunctionConnection, shall indicate the status and associated information about the 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 BACnetSCConnectionState, shall indicate the current state of the hub connection. Idle – The last connection was successfully disconnected. Connected – The connection is active. Failed - The last connection disconnected with errors.
Connector VMAC	This optional field, of type OCTET STRING (SIZE(6)), is the VMAC address of the connecting node. If the connecting node's VMAC has not been determined, this field shall be absent.
Connector UUID	This optional field, of type OCTET STRING (SIZE(16)), is the device UUID of the connecting node. If the connecting node's Device UUID has not been determined, this field shall be absent.
Error	This optional field, of type Error, shall indicate the most recent error for the connection. This field shall not represent any error reported through a BVLC-Result NAK message received through the connection. If the 'Connection State' field is Idle or Connected, this field shall be absent.
Details	This optional field, of type CharacterString, may be used to indicate details on the most recent error. If the 'Connection State' field is Idle or Connected, this field shall be absent.

12.56.Y17 SC_Direct_Connect_Accept_URIs

This property, of type BACnetARRAY[N] 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. 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 if the port supports accepting direct connections.

If this property is not configured, it shall contain an empty array.

12.56.Y18 SC_Direct_Connect_Connection_Status

This property, of type BACnetARRAY[N] of BACnetSCDirectConnect, shall indicate the status and associated information about the BACnet/SC direct connections initiated and accepted by this port.

If 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.

If the port is accepting a direct connection, an entry shall be added on or before the BACnet/SC connection has been established. It is a local matter how long the entry remains after disconnection.

The BACnetSCDirectConnect structure has the following fields:

URI	<p>This field, of type CharacterString, shall indicate the WebSocket URI for the direct connection.</p> <p>For initiated direct connections, this field shall contain the WebSocket URI that was used for initiating the connection.</p> <p>For accepted direct connections, this field shall be an empty string.</p>
Connection State	<p>This field, of type BACnetSCConnectionState, shall indicate the current state of the direct connection.</p> <p>Idle – The last connection was successfully disconnected. Connected – The connection is active. Failed - The last connection disconnected with errors.</p>
Peer VMAC	<p>This optional field, of type OCTET STRING (SIZE(6)), is the VMAC address of the connection peer node.</p> <p>If the connection peer node's VMAC has not been determined, this field shall be absent.</p>
Peer UUID	<p>This optional field, of type OCTET STRING (SIZE(16)), is the device UUID of the connection peer node.</p> <p>If the peer node Device UUID has not been determined, this field shall be absent.</p>
Error	<p>This optional field, of type Error, shall indicate the most recent error for the connection. This field shall not represent any error reported through a BVLC-Result NAK message received through the connection. If the 'Connection State' field is Idle or Connected, this field shall be absent.</p>
Details	<p>This optional field, of type CharacterString, may be used to indicate details on the most recent error. If the 'Connection State' field is Idle or Connected, this field shall be absent.</p>

12.56.Y19 SC_Reference_Ports

This property, of type BACnetLIST of UNSIGNED, contains a list of Network Port object instances that this port is allowed to use to accept or initiate WebSocket connections. An empty list indicates any valid Network Port object can be used. This property is not part of Network Port object hierarchy.

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.Y20 Operational_Certificate

This property, of type `OCTET STRING`, shall represent the X.509 certificate used to identify the IP host of the port.

The value of this property shall be an X.509 certificate in binary DER format. If no operational certificate is configured, this property, if present, shall be a zero length `OCTET STRING`.

This property shall support a certificate up to 2048 octets in size at a minimum. Support of larger certificates is optional.

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.Y21 CA_Certificates

This property, of type `BACnetARRAY[N]` of `OCTET STRING`, shall represent the certificate authorities for X.509 certificates the device accepts when it is establishing a secure connection.

Each `OCTET STRING` shall be an X.509 certificate in binary DER format. If no CA certificate is present in an array element, the `OCTET STRING` shall be of zero length.

This property shall support certificates up to 2048 octets in size in each element and shall support at least one entry. Support of larger certificates as well as multiple CA certificates is optional.

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 Private_Key

This write-only property, of type `BACnetARRAY[N]` of `OCTET STRING`, shall represent the private key related to the operational certificates present in the `Operational_Certificate` property.

The value of this property shall be a private key in PKCS #8 binary DER format (RFC 5958).

If this property is writable, writing a zero length `OCTET STRING` shall delete the private key.

Attempts to read this property shall return a `Result(-)` with an 'Error Class' of `PROPERTY` and an 'Error Code' of `READ_ACCESS_DENIED`.

If the port supports only an internally preset or generated private key and supports the `Certificate_Signing_Request` and `Certificate_Signing_Parameters` properties, then a write attempt to this property shall return a `Result(-)` with an 'Error Class' of `PROPERTY` and an 'Error Code' of `WRITE_ACCESS_DENIED`.

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.Y23 Certificate_Signing_Parameters

This property, of type BACnetCertificateSigningParameters, contains the signing parameters to be incorporated into the certificate presented in the Certificate_Signing_Request property. A write to this property shall initiate an internal mechanism to generate a certificate signing request.

The signing parameters are defined as follows:

Common Name	This parameter, of type CharacterString, shall indicate the common name (CN) to be contained in the signed certificate.
Country Name	This parameter, of type CharacterString, shall indicate the name of the country (C) to be contained in the signed certificate.
State or Province Name	This parameter, of type CharacterString, shall indicate the name of state or province (ST) to be contained in the signed certificate.
Locality Name	This parameter, of type CharacterString, shall indicate the name of the locality (L) to be contained in the signed certificate.
Organization Name	This parameter, of type CharacterString, shall indicate the name of the organization (O) to be contained in the signed certificate.
Organizational Unit Name	This optional parameter, of type CharacterString, shall indicate the name of the organizational unit (OU) to be contained in the signed certificate.
Email Address	This optional parameter, of type CharacterString, shall indicate the email address to contact the organization. Usually the email address of the certificate administrator or IT department.
Alternative Names	This optional parameter, of type SEQUENCE OF CharacterString, shall indicate the alternative names to be contained in the 'subjectAltName' field in the signed certificate.

If this property is not configured, then all parameters shall be empty character strings, and optional parameters shall be absent.

12.56.Y24 Certificate_Signing_Request

This read-only property, of type OCTET STRING, shall represent the PKCS #10 certificate signing request as defined by RFC 5967 that can be used by a certificate authority to create an operational certificate, matching an internally held private key.

The certificate presented in this property shall contain the parameters configured in the Certificate_Signing_Parameters property. The public keys and other parameters contained in this certificate shall be related to an internal private key. This private key is not exposed and cannot be configured from any remote entity. This private key is expected to be pre-assigned or generated and stored by some secure internal function of the device.

The certificate presented in this property shall be updated only when the Certificate_Signing_Parameters property is written. Whether the device is generating a new internal private key and public key for the updated certificate is a local matter.

The certificate presented in this property allows some external entity to initiate a certificate signing request to a CA to create an operational certificate signed by the CA. The selection of the signing CA is a site local matter.

If no certificate signing request is available, this property, if present, shall be a zero length OCTET STRING.

The resulting operational certificate signed by the CA can then be used by the external entity to configure the Operational Certificate property and will match the port's internal private key. The Private_Key property shall not be written in this case.

[Clause 21 New Productions]

[Add new ASN.1 productions to **Clause 21** section "**Base Types**" maintaining the alphabetical order, p. 805]

```
BACnetSCConnectionState ::= ENUMERATED {  
    idle             (0),  
    connected        (1),  
    failed           (2)  
}
```

```
BACnetSCDirectConnect ::= SEQUENCE {  
    uri                [0] CharacterString,  
    connection-state  [1] BACnetSCConnectionState OPTIONAL,  
    peer-vmac         [2] OCTET STRING (SIZE(6)) OPTIONAL,  
    peer-uuid         [3] OCTET STRING (SIZE(16)) OPTIONAL,  
    peer-hub-connector-state [4] BACnetSCHubConnectorState OPTIONAL,  
    error             [5] Error OPTIONAL,  
    details           [6] CharacterString OPTIONAL  
}
```

```
BACnetSCHubConnection ::= SEQUENCE {  
    connection-state [0] BACnetSCConnectionState,  
    error            [1] Error OPTIONAL,  
    details          [2] CharacterString OPTIONAL  
}
```

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

```
BACnetSCHubFunctionConnection ::= SEQUENCE {  
    connection-state [0] BACnetSCConnectionState,  
    connector-vmac   [1] OCTET STRING (SIZE(6)) OPTIONAL,  
    connector-uuid   [2] OCTET STRING (SIZE(16)) OPTIONAL,  
    error            [3] Error OPTIONAL,  
    details          [4] CharacterString OPTIONAL  
}
```

```
BACnetCertificateSigningParameters ::= SEQUENCE {  
    common-name       [0] CharacterString,  
    country-name      [1] CharacterString,  
    state-or-province-name [2] CharacterString,  
    locality-name     [3] CharacterString,  
    organization-name [4] CharacterString,  
    organizational-unit-name [5] CharacterString OPTIONAL,  
    email-address     [6] CharacterString OPTIONAL,  
    alternative-names [7] SEQUENCE OF CharacterString OPTIONAL  
}
```


[Clause 21 Changes to Existing Productions]

[Change **BACnetNetworkType** in **Clause 21**, p. 836]

```
BACnetNetworkType ::= ENUMERATED {  
    ...  
    serial          (10),  
    secure-connect (?),  
    ...  
}
```

-- 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.

[Insert into production **BACnetPropertyIdentifier** in **Clause 21**, preserving the alphabetical and numerical order, p. 845]

```
...  
ca-certificates          (?),  
certificate-signing-parameters (?),  
certificate-signing-request (?),  
max-bvlc-length-accepted (?),  
max-mpdu-length-accepted (?),  
operational-certificate (?),  
private-key              (?),  
sc-accepting-heartbeat-timeout (?),  
sc-connection-wait-timeout (?),  
sc-direct-connect-accept-uris (?),  
sc-direct-connect-connection-status (?),  
sc-disconnection-wait-timeout (?),  
sc-failover-hub-status (?),  
sc-failover-hub-uri (?),  
sc-hub-connector-state (?),  
sc-hub-function-accept-uris (?),  
sc-hub-function-connection-status (?),  
sc-hub-function-enable (?),  
sc-initiating-heartbeat-timeout (?),  
sc-primary-hub-status (?),  
sc-primary-hub-uri (?),  
sc-maximum-reconnect-time (?),  
sc-minimum-reconnect-time (?),  
sc-reference-ports      (?),  
...  
-- numerical order reference  
...  
-- see ca-certificates          (?),  
-- see certificate-signing-parameters (?),  
-- see certificate-signing-request (?),  
-- see max-bvlc-length-accepted (?),  
-- see max-mpdu-length-accepted (?),  
-- see operational-certificate (?),  
-- see private-key              (?),  
-- see sc-accepting-heartbeat-timeout (?),  
-- see sc-connection-wait-timeout (?),  
-- see sc-direct-connect-accept-uris (?),
```

-- see *sc-direct-connect-connection-status* (?),
-- see *sc-disconnection-wait-timeout* (?),
-- see *sc-failover-hub-status* (?),
-- see *sc-failover-hub-uri* (?),
-- see *sc-hub-connector-state* (?),
-- see *sc-hub-function-accept-uris* (?),
-- see *sc-hub-function-connection-status* (?),
-- see *sc-hub-function-enable* (?),
-- see *sc-initiating-heartbeat-timeout* (?),
-- see *sc-primary-hub-status* (?),
-- see *sc-primary-hub-uri* (?),
-- see *sc-maximum-reconnect-time* (?),
-- see *sc-minimum-reconnect-time* (?),
-- see *sc-reference-ports* (?),

...
}

-- The special property identifiers ...

[Insert into production **BACnetPropertyStates** in **Clause 21**, p. 862]

sc-connection-state [?] *BACnetSCConnectionState*,
sc-hub-connector-state [?] *BACnetSCHubConnectorState*,

[Change Addendum 135-2016bj Clause YY.1.8]

[Note to reviewer, the ‘x’ in the following refers to the value of the protocol revision of this addendum when published]

YY.1.8 BACnet/SC Network Port Objects

Participation in a BACnet/SC network is represented by single network port regardless of the number of connections and initiated or accepted WebSocket connections in use by the network port.

For BACnet/SC network port implementations less than protocol revision 17, the configuration of BACnet/SC network ports is a local matter and cannot be represented by Network Port objects.

For BACnet/SC network port implementations with a protocol revision 17 ~~and higher~~ *through x-1*, 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 required standard properties to be present see Clause 12.56.

For BACnet/SC network port implementations with a protocol revision x 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.

[Annex A PICS Changes for BACnet/SC]

[Change Annex A, p. 936]

ANNEX A - PROTOCOL IMPLEMENTATION CONFORMANCE STATEMENT (NORMATIVE)

...

Network Security Options:

If this product implements BACnet/SC or otherwise uses PKI certificates and keys, describe the methods and protocols used for their provision and management:

- ~~Non-secure Device is capable of operating without BACnet Network Security~~
- ~~Secure Device is capable of using BACnet Network Security (NS-SD-BIBB)~~
 - ~~Multiple Application-Specific Keys~~
 - ~~Supports encryption (NS-ED-BIBB)~~
 - ~~Key Server (NS-KS-BIBB)~~

[Add a new entry to **History of Revisions**, p. 1364]

(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	X	Addendum <i>cc</i> to ANSI/ASHRAE 135-2016 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. Extend the Network Port Object Type for BACnet/SC