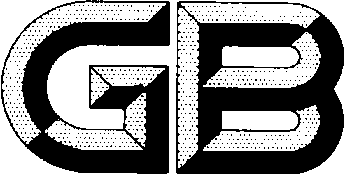
ICS  35.240.50

L 67



中华人民共和国国家标准

GB/T 28847.6—XXXX

|  |
| --- |
|  |

XXXX - XX - XX发布

XXXX - XX - XX实施

**国家市场监督管理总局**

**国家标准化管理委员会** **发布**

建筑自动化和控制系统

第6部分：数据通信协议一致性测试

Building automation and control systems —

Part 6: Data communication conformance testing

|  |
| --- |
| （征求意见稿） |
|  |

目 次

[前言 Ⅱ](#_Toc55414061)

[1 范围 1](#_Toc55414062)

[2 规范性引用文件 1](#_Toc55414063)

[3 术语和定义 2](#_Toc55414064)

[4 缩略语和符号 16](#_Toc55414065)

[5 协议实现一致性声明EPICS的文件格式 20](#_Toc55414066)

[6 协议实现一致性声明EPICS的测试内容 47](#_Toc55414067)

[7 一致性测试脚本语言TCSL的规定 48](#_Toc55414068)

[8 一致性测试所涉及的对象属性 56](#_Toc55414069)

[9 应用服务启动测试 165](#_Toc55414070)

[10 应用程序服务执行测试的一般规定 264](#_Toc55414071)

[11 网络层协议测试 424](#_Toc55414072)

[12 逻辑链路层协议测试 470](#_Toc55414073)

[13 数据链路层协议测试 473](#_Toc55414074)

[14 特殊功能测试 575](#_Toc55414075)

[15 IP功能测试 590](#_Toc55414076)

[16 报告测试结果 621](#_Toc55414077)

[附录A （资料性附录）电子协议实现一致性声明EPICS示例 622](#_Toc55414078)

前  言

GB/T 28847《建筑自动化和控制系统》分为七个部分：

1. 第1部分：概述；
2. 第2部分：硬件；
3. 第3部分：功能；
4. 第4部分：应用；
5. 第5部分：数据通信协议；
6. 第6部分：数据通信一致性测试；
7. 第7部分：工程实现。

本部分为GB/T 28847的第6部分。

本部分按照GB/T 1.1－2020给出的规则起草。

本部分由中华人民共和国住房和城乡建设部提出。

本部分由全国智能建筑及居住区数字化标准化技术委员会（SAC/TC 426）归口。

本部分起草单位：

本部分主要起草人：

建筑自动化和控制系统 第6部分:数据通信一致性测试

1. 范围

GB/T 28847.6部分规定了建筑自动化和控制系统中数据通信一致性测试所涉及的内容，包括协议实现一致性声明EPICS的文件格式、协议实现一致性声明EPICS的测试内容、一致性测试脚本语言TCSL的规定、一致性测试所涉及的对象属性、应用服务启动测试、应用服务执行测试、网络层协议测试、逻辑链路层协议测试、数据链路层协议测试、特殊功能测试、IP功能测试和报告测试结果等。

本部分适用于建筑自动化和控制系统中相关设备之间数据通信的协议一致性测试。

1. 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注明日期的引用文件，仅注明日期的版本适用于本文件。凡是不注明日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

ISO 7498（所有部分）信息处理系统 开放系统互连 基本参见模型（Information processing systems—Open systems interconnection—Basic reference model）

ISO TR 8509 信息处理系统 开放系统互连 服务协定（Information processing systems—Open Systems Interconnection—Service conventions）

ISO/IEC 8802-2 信息处理系统 局域网 第2部分：逻辑链路控制（Information processing systems—Local area networks—Part 2: Logical link control）

ISO/IEC 8802-3 信息处理系统 局域网 第3部分：载波侦听多路访问冲突检测(CSMA/CD）访问方法和物理层规范[Information processing systems—Local area networks—Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications]

ISO/IEC 8824 信息技术 开放系统互连 抽象语法记法.1(ASN.1)：基本记法规范[Information technology—Open systems interconnection—Specification of abstract syntax notation one (ASN.1)]

ISO/IEC 8825 信息技术 开放系统互连 抽象语法记法.1(ASN.1)：基本编码规则规范[Information technology—Open systems interconnection—Specification of basic encoding rules for abstract syntax notation one (ASN.1)]

1. 术语和定义

下列术语和定义适用于本文件。



抽象语法 abstract syntax

应用层数据或应用协议控制信息使用独立于表示它们的编码技术的符号规则要遵循的规定。



应用 application

用户信息处理请求的集合。



应用实体 application-entity

一个应用进程中的与OSI相关的部分。



应用进程 application-process

实开放系统中一个特定应用进行信息处理的元素。



应用协议控制信息 application-protocol-control-information

使用表示层服务来协调连接操作的应用实体间交换信息。



应用协议数据单元 application-protocol-data-unit

应用协议中规定的数据单元，包含应用协议控制信息等。



应用服务元素 application-service-element

在恰当的时候，通过使用下层服务提供OSI环境，应用实体中的一部分。



具体语法 concrete syntax

某数据的特定表示方式的具体实现，用于数据的正式规范中的语法规则。



确认（原语） confirm （primitive）

一种交互的表示，在该交互中一个服务提供者指示在一个特定的服务访问点上某些规程已经完成。



指示（原语） indication （primitive）

一种交互的表示，在该交互中服务提供者要么指示它主动请求了一些规程，要么指示一个规程已经被位于对等服务访问点的服务用户所请求。



对等实体 peer-entities

位于协议中同一层的实体。



实开放系统 real open system

在与其它实系统通信时遵守OSI标准的要求的一个实系统。



实系统 real system

由一台或多台电脑、相关软件、外部软件、终端、操作员、物理过程、信息传递方法等组成的集合。注：是一个自主的整体，具有信息处理和信息传递的能力。



请求（原语） request（primitive）

在交互中服务用户请求执行一些规程的一种交互的表示。



响应（原语） response（primitive）

在交互中服务用户指示它已经完成了一些规程的一种交互的表示。



第N层服务访问点 （N）-service-access-point

第N层实体向第N-1层实体提供第N层服务的访问点。



第N层服务数据单元 （N）-service-data-unit

第N层的接口数据，在第N层通信中保持一致性。



服务用户 service-user

单一开放系统中一个通过服务访问点使用服务的实体。



服务原语 service-primitive

服务用户和服务提供者之间交互一种抽象的、与具体实现无关的表示。



服务提供者 service-provider

向对等服务用户提供服务的所有实体的抽象。



传输语法 transfer-syntax

在开放系统间进行数据传输所使用的具体语法。



用户元素 user element

为了实现应用进程的通信目的而使用所需的应用服务元素的一个应用进程的某部分的表示。



访问控制 access control

调整和限制访问网络资源的方法。



访问权限（物理访问控制） access rights（physical access control）

给予一个证书的访问特权。



访问用户（物理访问控制） access user （physical access control）

持有一个或多个证书的人或装置。



警报 alarm

以声音或者视觉或者两者都有的形式，提醒人类操作员发生了一个可能需要校正的异常工况的一个通告。



警报确认 alarm-acknowledgment

指示一个人类操作员已经查看并对事件通知做出反应的进程。



算法改变报告 algorithmic change reporting

基于一个事件登记（Event Enrollment）对象中指定的算法，探测和报告一个警报或事件。



鉴别 authentication

验证身份的行为。



鉴别因子 authentication factor

用于证明证书身份的数据元素。



鉴别（网络安全） authorization （network security）

基于已知身份和访问规则，对网络资源的访问进行控制。



鉴别（物理访问控制） authorization （physical access control）

决定某访问用户是否被允许通过一个访问控制点进入某个受保护区域的进程。



BACnet设备 BACnet device

支持使用BACnet协议进行数字通信的一个现实的或虚拟的设备。



BACnet 用户 BACnet-user

用BACnet用户元素来表示应用进程的一部分。



闪烁警报 blink-warn

照明控制中的一种典型方法，用以通知房间居住者灯光即将被自动关闭。

在该过程中，灯光可能闪烁一次或多次，或者产生一个声音信号。警报发生后，房间照明会维持一段时间，以便居住者要么安全离开房间，要么发起一个维持房间照明的请求。也被称为“flick 警报”或“闪光警报”。



网桥 bridge

在物理层或数据链路层连接两个以上网段的设备。

该设备可基于MAC层地址实行报文过滤。



广播 broadcast

作为一个单一单元被应用到多个设备的报文。



状态改变 change of state

当测试或计算出的布尔值或离散枚举值发生改变时发生的事件。



值的改变 change of value

当测试或计算出的模拟值被一个预定义的值改变时发生的事件。



客户 client

一个系统或设备，为了某些特殊目的，通过一个服务请求实例使用另一个设备。

客户向服务器请求服务。



控制器 controller

调整或管理系统或部件的设备。



证书（物理访问控制） credential （physical access control）

鉴别因子和访问权限的组合。



数据机密性 data confidentiality

对非法个体、实体或者进程，信息不可获取或不可公开的性质。



数据完整性 data integrity

数据不能被非法方式改变或破坏的性质。



数据来源鉴别 data origin authentication

对收到的数据的来源是否为其声明的来源的证实行为。



日期模式 date pattern

包含一个以上未指定字节或者特定日期值的日期。



直连网络 directly connected network

可通过路由器直接访问，不需中间路由器转发报文的网络。

当PTP连接有效并且没有使用中间路由器时，该PTP连接是直连网络。



下载 download

一种特定类型的文件传输，将可执行程序或者数据库传送到一个远程设备，在该远程设备上这些程序或数据库可以被执行。



密文 encrypted message

通过密码系统产生的不可直接理解的文字或信号。



实体 entity

可以被一组属性描述的可辨认的、具有独立的和可区分特征的存在事物。



错误探测 error detection

用于鉴定通信过程中是否出现错误的规程。



错误恢复 error recovery

由探测到的错误激活的规程，用于确保信息交换得以继续。



事件算法 event algorithm

用于决定事件-初始化（event-initiating）对象切换正常和非正常状态的算法。



事件-初始化对象 event-initiating object

可以用于监视其自身状态并报告事件状态的变化。



事件-通知-分配 event-notification-distribution

当对象改变了事件或者确认状态，服务器执行决定通知客户和发送信息。



事件通知报文 event notification message

用于指示在一个事件或者一个事件-初始化对象的确认状态中改变的ConfirmedEventNotification或者 UnconfirmedEventNotification服务请求。



事件-状态-探测 event-state-detection

执行对象事件算法，并监视对象属性的进程。



事件概要 event-summarization

通过其中一个事件概要服务查询设备中的事件-初始化对象，以决定哪些事件-初始化对象满足特定事件状态或报告条件。



衰减 fading

实际输出在一定周期内从一个设置值逐渐增加或减少到另一个值。



网关 gateway

连接两个以上不同网络的设备，从而允许信息在这些网络间进行交换。



全局 global

通信网络上的所有设备或节点。



全局广播 global broadcast

被发送到一个BACnet互联网上所有网络中的或节点的报文。



半路由器 half router

PTP连接的一端的设备或节点。



无安全能力设备 incapable device

本质上没有能力，或者被配置为没有能力产生和消耗安全BACnet报文的设备。



初始化 initialization

由上电引起的建立一个已知状态的过程。



互联网络 internetwork

路由器连接的两个以上网络组成的集合。



内部报告 intrinsic reporting

警报或事件的探测和报告，基于对象类型规定中的算法。不涉及对Event Enrollment对象的外部引用。



反网络 inverted network

BACnet互联网中有两个以上的网络被一个网络连接，相关的NPDU大小小于其加入的网络。



密钥 key

控制加密和解密操作的一个符号序列。



本地 local

同一网络上作为参见设备的相关设备。



本地广播 local broadcast

被发送到同一网络上所有设备或节点的报文。



本地网络 local network

与BACnet设备直接连接的网络。



介质 medium

物理传输实体。双绞线、光纤和同轴电缆是典型的介质。



介质访问控制 medium access control

维持对通信介质的访问次序的过程。



网络 network

由网桥连接起来的网络地址相同的一个以上网段组成的集合。



网络资源 network resource

任何可以通过一个通信介质访问的物理或逻辑实体。



节点 node

连接到通信介质的可寻址设备。



通知-客户 notification-client

接收和处理事件通知报文的BACnet设备。



通知-服务器 notification-server

包含事件-初始化对象并执行事件通知分发的BACnet设备。



对象 object

数据分类的特定实例。



对象配置 object profile

配置定义了一个由属性、行为、和/或对一个私有对象的要求、或对一个标准对象的私有扩展组成的集合。



对象类型 object type

数据的分类。



操作员鉴别 operator authentication

对登陆到设备上的操作员是否为其声明身份的证实行为。



对等实体鉴别 peer entity authentication

对联合体中一个对等实体是否为其声明身份的证实行为。



物理访问控制 physical access control （PACS）

对保护区域进行控制的电子系统。



物理网段 physical segment

BACnet节点附于其上的一个单一连续介质。



物理安全 physically secure

有阻止未授权个体进行物理访问能力的设备或网络。



普通设备 plain device

无安全能力的设备。



普通网络 plain network

不要求签名或加密通信的网络。



普通报文 plain message

未被BACnet安全封装保护的一个报文。



可打印字符 printable character

与设备控制字符形成对照，代表一个可打印符号的字符。



属性 property

对象类型的一个特定特性。



私有 proprietary

在BACnet的范围内确定的对象类型、属性、私有传输服务或枚举的任何扩充或附加部分。



加速 ramping

实际输出以固定的变化速度从一个设定值逐渐增加或减少到另一个值。



接收BACnet用户 receiving BACnet-user

接收指示或证实服务原语的BACnet用户。



可靠性评估 reliability-evaluation

进程自身可靠性和用于设置可靠性属性的值。



远程 remote

位于和参见节点不同网络的相关设备。



远程广播 remote broadcast

被发送到与源不同网络上所有设备和节点的报文。



远程网络 remote network

必须通过一个或多个路由器才能访问BACnet设备的网络。



中继器 repeater

在物理层上连接两个以上物理网段的设备



请求BACnet用户 requesting BACnet-user

在有证实服务中作为客户的BACnet用户。



响应BACnet用户 responding BACnet-user

在有证实服务中作为服务器的BACnet用户。



基于角色的访问控制 role-based access control

分配给特定角色的访问权限。

注：访问用户通过他们被分配的角色获取权限。



路由器 router

在网络层连接两个以上网络的设备。



安全网络 secure network

所有通信都要被签名和加密的网络。



安全 security

保护信息交换，防止信息被泄露给非法个体的规程。

注：安全措施甚至用来阻止敏感信息泄露给那些能合法访问通信网络的个体。安全与访问控制明显不同，即使有些安全可以通过限制对通信介质的物理访问来实现。



网段 segment

包含用中继器连接的一个以上物理网段的网段。



发送BACnet用户 sending BACnet-user

发送一个请求或响应服务原语的BACnet用户。



服务器 server

响应为某些特定目的的服务请求实例的一个系统或设备。



签名报文 signed message

被安全头封装、签名但没有被加密的报文。



特定日期 specific date

完全规定的日期。

注：如January 24, 1991, Day of week=Thursday。一个特定日期不包含未规定字节或特殊日期值。



规定日期时间 specific datetime

由一个规定日期和一个规定时间组成的一个BACnetDateTime。



规定时间 specific time

完全规定的时间，不包含未规定字节。

注：例如，17:35:45.17（=5:35:45.17 P.M.）。



标准对象类型 standard object type

数值取值在保留范围。



标准属性 standard property

标准对象类型的特性。



步进 stepping

离散数值的输出值的增加或减少。



同步 synchronization

允许进程在传输或交换中定义和鉴别某些特定位置的功能，这些位置用于将某次通信会话重置为一个预先设置的状态。



时间模式 time pattern

可能包含一个以上未规定字节的时间。



时间戳 timestamp

指示伴随事件或操作记录的时间记录中的一个点。



单位时间 unit time

传送一个带有一个比特的发送位和一个比特停止位的字节所需的时间。



未规定日期 unspecified date

完全由未规定字节组成的日期（X“FF”的值= D“255”）。



未规定日期时间 unspecified datetime

由一个未规定日期和一个未规定时间组成的BACnetDateTime。



未规定字节 unspecified octet

用在包含值X“FF”=D“255”的日期、时间或BACnetWeekNDay值中的字节。



未规定时间 unspecified time

完全由未规定字节组成的时间（X“FF”的值=D“255”）。



上传 upload

从一个远程设备传输一个可执行程序或数据库的过程。



测试数据库 test database

即BACnet功能和对象的数据库，该数据库通过读取EPICS的内容创建。

1. 缩略语和符号
   1. 缩略语

下列缩略语适用于本文件。

AE：应用实体（Application Entity）

APCI：应用协议控制信息（Application Protocol Control Information）

APDU：应用层协议数据单元（Application Layer Protocol Data Unit）

API：应用程序接口（Application Program Interface）

ARCNET：附加资源计算机网络（Attached Resource Computer Network）

ASE：应用服务单元（Application Service Element）

ASN.1：抽象语法表示法1（Abstract Syntax Notation One)

BAC：建筑自动化和控制（Building Automation and Control）

BBMD：BACnet/IP广播管理设备（BACnet/IP Broadcast Management Device）

BDT：广播分布表（Broadcast Distribution Table）

BIBBs：BACnet互操作组成模块（BACnet Interoperability Building Blocks）

B/IP：BACnet/IP（BACnet/IP）

B/IP-M：BACnet/IP多播（BACnet/IP Multicast）

BNF：巴科斯范式语法（Backus-Naur form）

BVLC：BACnet虚拟链路控制（BACnet Virtual Link Control）

BVLCI：BACnet虚拟链路控制信息（BACnet Virtual Link Control Information）

BVLL：BACnet虚拟链路层（BACnet Virtual Link Layer）

CNF：确认原语（Confirm Primitive）

COV：值的改变（Change of Value）

CRC：循环冗余校验（Cyclic Redundancy Check）

DA：本地目标MAC层地址（Local Destination MAC Layer Address）

DADR：最终目标MAC层地址（Ultimate Destination MAC Layer Address）

DER：需要应答的数据（Data Expecting Reply）

DES：数据加密标准（FIPS 46-1）[Data Encryption Standard (FIPS 46-1)]

DESFire：数据加密标准快速，创新，可靠和安全（Data Encryption Standard Fast, Innovative, Reliable and Secure）

DLEN：最终目标MAC层地址的1个字节长度（1-octet Length of Ultimate Destination MAC Layer Address）

DNET：2字节的最终目标网络号（2-octet Ultimate Destination Network Number）

DSAP：LLC目标服务访问点 （对于BACnet为X“82”）[LLC Destination Service Access Point (X“82”for BACnet)]

EIB：欧洲安装总线（European Installation Bus）

EPICS：电子协议实现一致性声明（Protocol Implementation Conformance Statement）

EXEC：执行一个服务请求的能力（Capable of Executing a Service Request）

FDT：外部设备表（Foreign Device Table）

ICI：接口控制信息（Interface Control Information）

IL：ARCNET信息长度字节（ARCNET Information Length Field）

IND：指示原语（Indication Primitive）

INF：“无穷大”, 代表正无穷大的一个独一无二的二进制模型（“Infinity”, a Unique Binary Pattern Representing Positive Infinity）

-INF：“负无穷大”, 代表负正无穷大的一个独一无二的二进制模型 （“Negative infinity”, a Unique Binary Pattern Representing Negative Infinity）

INIT：发起一个服务请求的能力（Capable of Initiating a Service Request）

IP：互联网协议RFC 791（Internet Protocol，RFC 791）

IUT: 测试的实现（implementation under test）

KNX：Konnex系统规范（The Konnex System Specification）

LAN：局域网（Local Area Network）

LLC：逻辑链路控制（Logical Link Control)

LPCI：链路协议控制信息（Link Protocol Control Information）

LPDU：链路协议数据单元（Link Protocol Data Unit）

LRC：纵向冗余校验（Longitudinal Redundancy Check）

LSAP：链路服务访问点（对于BACnet为X“82”）[Link Service Access Point (X“82” for BACnet)]

LSDU：链路服务数据单元（Link Service Data Unit）

MA：介质访问（前缀）[Medium Access (Prefix)]

MAC：介质访问控制（Medium Access Control）

MPCI：MAC协议控制信息（MAC Protocol Control Information）

MPDU：MAC层协议数据单元（MAC Layer Protocol Data Unit）

MSDU：MAC服务数据单元（MAC Service Data Unit）

MS/TP：主从/令牌传送（Master-Slave/Token-Passing）

NaN：“非数字”，代表一个无效数字的独一无二的二进制模型（“Not a Number”, a Unique Binary Pattern Representing an Invalid Number）

NAT：网络地址转换RFC 2663（Network Address Translation，RFC 2663）

NP：网络优先级（Network Priority）

NPCI：网络协议控制信息（Network Protocol Control Information）

NPDU：网络层协议数据单元（Network Layer Protocol Data Unit）

NRZ：不归零制（Non-return to Zero）

NSAP：网络服务访问点（Network Service Access Point）

NSDU：网络服务数据单元（Network Service Data Unit）

OSI：开放系统互连（Open Systems Interconnection）

PAC：ARCNET数据包的头字节（ARCNET Data Packet Header Octet）

PCI：协议控制信息（Protocol Control Information）

PDU：协议数据单元（Protocol Data Unit）

PICS：协议实现一致性声明（Protocol Implementation Conformance Statement）

PK：私钥（Private Key）

PPCI：物理层协议控制协议（Physical Layer Protocol Control Information）

PPDU：物理协议数据单元（Physical Protocol Data Unit）

PPP：点对点协议RFC 1661（Point-To-Point Protocol，RFC 1661）

PSDU：物理服务数据单元（Physical Service Data Unit）

PTP：点对点（Point-To-Point）

REQ：请求原语（Request Primitive）

RFC：请求注解（Request for Comment）

RSP：响应原语（Response Primitive）

SA：本地网络源MAC层地址（Local Network Source MAC Layer Address）

SAP：服务访问点（Service Access Point）

SC：ARCNET系统码（对于BACnet为X“CD”）[ARCNET System Code (X“CD” for BACnet)]

SDU：服务数据单元（Service Data Unit）

SIA：安全工业协会（Security Industry Association）

SID：ARCNET源MAC地址（ARCNET Source MAC Address）

SK：会话密钥（Session Key）

SLEN：源网络MAC层地址的1个字节长度（1-octet Length of Original Source MAC Layer Address）

SLIP：串行线路IP协议RFC 1055（Serial Line Internet Protocol，RFC 1055）

SNET：2字节表示的初始源网络号（2-octet Original Source Network Number）

SPC：标准工程委员会（Standard Project Committee）

SSAP：LLC源服务访问点（对于BACnet为X“82”）[LLC Source Service Access Point (X“82” for BACnet)]

TCSL：测试一致性脚本语言（Test Conformance Script Language）

TD：测试设备（Test Devices）

TPI：文本协议信息（Text Protocol information）

TSM：事务状态机（Transaction State Machine）

UART：通用异步接收器/发射器（Universal Asynchronous Receiver/Transmitter）

UDP：用户数据报协议RFC 768（User Datagram Protocol，RFC 768）

UTC：协调世界时（Universal Time Coordinated）

VT：虚拟终端（Virtual Terminal）

XID：交换标识（Exchange Identification）

* 1. 符号

下列符号适用于本文件。

A：应用层（前缀）[Application Layer (Prefix)]

B“” ：在双引号之间使用二进制（Denotes that Binary Notation is Used between the Single Quotes）

C：有条件的（Conditional）

C（=）：条件（在语义上，参数与服务原语中左边的参数相等）[Conditional (The Parameter is Semantically Equivalent to the Parameter in the Service Primitive to Its Immediate Left in the Table)]

D“”：在双引号之间使用十进制（Denotes that Decimal Notation is Used between the Single Quotes）

L：数据链路（前缀）[Data Link (Prefix)]

M：命令的（Mandatory）

M（=）：命令的（在语义上，参数与服务原语中左边的参数相等）[Mandatory (The Parameter is Semantically Equivalent to the Parameter in the Service Primitive to Its Immediate Left in the Table)]

N：网络层（前缀）[Network Layer (Prefix)]

O：对该属性的支持是可选的（Support of a Property is Optional）

P：物理层（前缀）[Physical Layer (Prefix)]

R：BACnet服务支持和可读属性（A Property shall be Supported and Readable Using BACnet Services）

S：选择（Selection）

S（=）：选择（在语义上，参数与服务原语中左边的参数相等）[Selection (The Parameter is Semantically Equivalent to the Parameter in the Service Primitive to Its Immediate Left in the Table)]

U：用户选项（User Option）

U（=）：用户选项（在语义上，参数与服务原语中左边的参数相等）[User Option (The Parameter is Semantically Equivalent to the Parameter in the Service Primitive to its Immediate Left in the Table)]

W：BACnet服务支持，可读和可写属性（A Property shall be Supported, Readable, and Writable using BACnet Services）

X“”：在双引号之间使用十六进制（Hexadecimal Notation is Used between the Single Quotes）

1. 协议实现一致性声明EPICS的文件格式

电子协议实现一致性声明（EPICS）文件包含以标准文本形式表示的BACnet协议实现一致性声明。EPICS文件是对指定设备BACnet对象和服务执行测试的文件，可被机器和人读取。EPICS文件需要使用扩展名“.TPI”（文本协议信息），由正常的可编辑的文本行构成，文本行由回车/换行符（X’0D’, X’0A’）结尾的字符代码组成。

软件测试工具用EPICS文件来执行和解释该标准中定义的测试结果。依据本标准流程进行测试的所有设备均随附EPICS文件。

* 1. 字符编码

BACnet提供多种可能的字符编码。BACnet协议中字符编码分为三组：单字节流（八位位组），双字节流和四字节流。单字节流的字符为单字节值。在某些情况下，如微软系统DBCS和JIS C 6226，用特殊的字节值表示其随后紧跟的字节与前面字节一起被视作单个字符，这样可将字符扩大为大于256的字符。而双字节流把双字节当做单个字符，例如ISO 10646 UCS-2编码，此时双字节中的首字节是最重要的部分。而四字节流，如ISO 10646 UCS-4，将四个字节作为单字符，同样四字节流中的首字节也是最重要的。

为了适用于BACnet设备声明中可能使用的各类编码，EPICS文件的头字节既能将文件识别为EPICS文件，也能识别特定的编码。头字节以字符串“PICS#”开头，#为表示字符集的数字，如表5-1所示。

**表5-1** 字符集代码

|  |  |
| --- | --- |
| 编码 | 字符集 |
| 0 | ANSI X3.4 |
| 1 | Microsoft DBCS |
| 2 | JIS C 6226 |
| 3 | ISO 10646(UCS-4) |
| 4 | ISO 10646(UCS-2) |
| 5 | ISO 8859-1 |

通过检查EPICS文件的前八个字节，可识别单字节流格式。以ANSI X3.4编码为例，这八个八位字节包含：X’50’ X’49’ X’43’ X’53’ X’20’ X’30’ X’0D’ X’0A’。后两个字节代表文本“PICS 0”后所跟的回车符和换行符。

双字节流格式通过检查EPICS文件的前十六个字节来识别。以ISO 10646 UCS-2编码为例，这十六个字节包含：

X’00’ X’50’ X’00’ X’49’ X’00’ X’43’ X’00’ X’53’

X’00’ X’20’ X’00’ X’34’ X’00’ X’0D’ X’00’ X’0A’

最后一行表示文本“PICS 4”后所跟的回车符和换行符。

四字节流格式通过检查EPICS文件的前三十二个字节来识别。以ISO 10646 UCS-4编码为例，这三十二个字节包含：

X’00’ X’00’ X’00’ X’50’ X’00’ X’00’ X’00’ X’49’

X’00’ X’00’ X’00’ X’43’ X’00’ X’00’ X’00’ X’53’

X’00’ X’00’ X’00’ X’20’ X’00’ X’00’ X’00’ X’33’

X’00’ X’00’ X’00’ X’0D’ X’00’ X’00’ X’00’ X’0A’

最后一行表示文本“PICS 3”后所跟的回车符和换行符。

* 1. 电子协议实现一致性声明EPICS文件结构

EPICS文件由以回车/换行对（X’0D’，X’0A’）结尾的文本行组成，文本行根据5.1中定义的字符编码被编为单字节流、双字节流或四字节流。本标准的其他部分，基于EPICS头文件的字符编码，“字符”表示一个符号以单、双或四字节来编码。例如，字符可以被编码为X’20’或X’0020’或X’00000020’。在此标准中，所有的字符将以单个字节（八位位组）形式显示。

本测试条款使用特殊符号**↵**表示一个回车/换行对（X’0D0A’）。在字符串外，字符代码标签（X’09’）、空格（X’20’），回车（X’0D’）和换行符（X’0A’）都被视作无效字符。包含有一个或多个无效字符的序列相当于单个无效字符。在字符串外，两个破折号（X’2D’）序列表示注释的开始，该注释以下一条回车/换行对作为结束，即该行结束后的显示符号--。注释被视作无效，可随意插入。

EPICS文件用以下文本作为首行：

**BACnet Protocol Implementation Conformance Statement ↵**

此行文本是识别EPICS文件格式的一个重要标志。

上述签名行后是定义EPICS部分(见5.5)的文本及用于实现给定设备特定数据的文本。

EPICS文件的最后一行为以下文本：

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

* 1. 字符串

双引号（X’22’），单引号（X’27’）和重音符（X’60’）的出现都代表字符串。对于双引号，一个字符串结束或者该行结束时要以下一个双引号出现作为字符串的结尾。对于单引号或重音符，字符串的结束或该行结束时要以下一个单引号（X’27’）出现作为字符串的结尾。因此，如果字符串中包括单引号或重音符的文字字符，需要使用双引号表示字符串的开始和结束；而字符中包括双引号的字符串应使用单引号或重音符来表示字符串的开始和结束。

* 1. 参数值的规则

参数有多种形式，不同部分的参数以不同形式表示。以下规则定义了参数格式：

（a）关键词不区分大小写，所以X’41’到X’5A’等效于X’61’到X’7A’；

（b）空值由字符串“NULL”表示；

（c）布尔值由字符串“T”或“TRUE”表示值正确，“F”或“FALSE”表示值错误；

（d）整数值显示为数字字符串，数字前面可能有负号（-）：12345或-111；

（e）实数值要显示小数点，但小数点不能在第一个或最后一个字符，例如值可以是1.23，0.02，1.0，但不能是.02；

（f）单字节八位字符串用单引号（X’2D’）或重音符（X'60'）括起来的十六进制数字对表示，且字符串前要加上字母“X”：X’001122’；

（g）字符串用一个或多个用单引号、重音符或双引号括起来的字符表示，如5.3所示：’text’或'text'或”text”；

（h）长字符串用列表表示，用大括号（{ }或X’7B’和X’7D’）括起来，真假值表示为{T，T，F} 或{TRUE, TRUE, FALSE}。当其实际的值不重要时，使用问号{T，T，?}；

（i）枚举值用指定值而非数字值表示。枚举名称不分大小写，所以X’41’到X’5A’相当于X’61’到X’7A’。枚举名称中下划线（X’5F’）和破折号（X’2D’）是等效的。专有值显示为没有空格的命名文本，并以非负十进制数字结尾。每个枚举值必须以“proprietary”开头：Object\_Type，proprietary-object-type-653；

（j）日期要用圆括号括起来：(Monday,24-January-1998)。通配符或未指定的字段用星号(X’2A’)表示:(Monday,\*-January-1998)。日期中未注明周几表示未指定：(24-January-1998)；

（k）时间用小时、分钟、秒表示，格式为hh:mm:ss.xx: 2:05:44.00,16:54:59.99。通配符段用星号（X’2A’）表示：16:54:\*.\*；

（l）对象标识符用括号括起来，用逗号分隔对象类型和实例数量，如(analog-input,56)。专有对象类型用“proprietary”替换对象类型枚举，其后是对象类型的数值，如(proprietary 700,1)；

（m）构造的数据项是由大括号（{ }或X’7B’和X’7D’）括起来，元素间用逗号隔开。如果其中的一个元素本身也是一个构造值，则该元素也要由大括号括起来。

5.4.1 复杂的参数值

一些参数值，特别是构造或者CHOICE类型的编码值的属性值，需要使用更复杂的符号来表示他们的值。对于这些属性值的表示方法与其ASN.1编码相关联，并且可能因上下文而模糊不清。因此用以下附加规则规定这些类型的参数值的表示形式：

（a）作为应用标记值的CHOICE值，应按照5.4中所述编码的选择项目的值来表示；

（b）带有上下文标记值的CHOICE值，用放在方括号内的上下文标记数值来表示，其后是所选择项目的值；

（c）列表值（ASN.1“SEQUENCE OF”）用括号括起来表示，列表中的元素用逗号分隔。如果一个元素本身是一个构造值，则该元素也要由大括号括起来；

（d）数组值用大括号括起来表示，数组中的元素用逗号分隔。如果一个元素本身是一个构造值，则该元素也要由大括号括起来。

5.4.2指定参数值范围

一些属性可能对其值的范围或分辨率有限制。为了检测测试结果中哪些属性的值发生改变，可以使用WriteProperty，WritePropertyMultiple或AddListElement，然后使用ReadProperty或ReadPropertyMultiple回读，因此有必要知道这些参数值的极限值。测试数据库可能包含定义这些约束的限制语句。它们允许适用的极限值和数据类型有：

（a）最小值-最小值为无符号、整数、实数，或双数据类型。是日期数据类型的最早日期；

（b）最大值-最大值为无符号、整数、实数，或双数据类型。是日期数据类型的最晚日期；

（c）分辨率-最低保证分辨率是实数和双数据类型。最小时间分辨是时间为秒的数据类型；

（d）最大长度字符串-字符串或八进制字符串的最大长度；

（e）最大长度列表-一个列表中保证的最大元素的最大数目；

（f）最大长度数组-数组中元素的最大数目；

（g）允许值-一个逗号分隔的枚举数据类型支持的枚举列表。引用外部对象标识符的属性的对象类型的逗号分隔列表。

限制性陈述须在默认值后在指明的括号内（<and>）列出。如果在一个单一的内角括号内有多个限制，那么这些限制应以分号（；）分隔。限制语句包含限定名后跟一个冒号（：），其次是限制值，或者在适当的情况下，一个用逗号分隔可能值的列表。

下面是一些属性值的例子，它们在测试数据库中可能出现：

present-value: 13.4 <minimum: 0.0; maximum: 20.0; resolution: 0.1>

description: "this is a description" <maximum length string: 30>

units: milliamperes <allowed values: milliamperes, amperes>

object-property-reference: (analog input, 12) <allowed values: analog input, analog value>

Units属性是特殊的情况，因为Units的改变可以改变Present\_Value属性的值以及对其值的任何限制。因此，最小值、最大值和分辨率的限制仅对Units属性的默认值有效。

如5.5.8所述，可以为大多数数据类型指定默认范围限制。测试数据库中的限制语句包含限制语句的单个属性的默认限制。

* 1. 文件内容

每一段EPICS文件开始的一节名称后跟一个冒号（：或X'3A'）。冒号后是一组一个或多个参数的大括号（{ }或X'7B'X'7D'）。

使用下列符号作为占位符来表示参数信息的存在：

（a）引号内的开放式符号，“**□**”，用于表示字符串参数；

（b）无引号的开放式符号，**□**，用来表示数据类型参数，而不是字符串的参数；

（c）一个问号，？，在测试数据库中用于表明该属性是存在的，但该值是未知的，因为它取决于硬件输入，或正在由内部算法更改。

EPICS文件的例子可以在附件A中查阅。

1. 一般信息部分

这部分提供了BACnet设备的一般信息。这些部分的语法如下：

Vendor Name: "**□**"**↵**

Product Name: "**□**"**↵**

Product Model Number: "**□**"**↵**

Product Description: "**□**"**↵**

1. 一致性部分

这部分提供了相关设备声称的支持BACnet功能的信息。

1. 支持的BACnet互操作组成模块BIBBs

本节说明哪些BIBB可被支持。语法如下，大括号内列出每个受支持的BIBB，每行一个。空列表表明不支持BIBBs。

BIBBs Supported: **↵**

{**↵**

**□↵**

}**↵**

BIBBs可以是附件K中所述的任何BIBBs。EPICS中的格式应使用每个BIBB标题中描述的缩写。例如：支持’Data Sharing - ReadProperty - B’的设备在此部分应使用’DS-RP-B’。

例如：

BIBBS Supported: **↵** 

{**↵**

DS-RP-B**↵**

DS-WP-B**↵**

DS-RPM-B**↵**

DM-DOB-B**↵**

DM-DDB-B**↵**

DM-DDB-A**↵**

}

1. 支持的应用服务

本部分说明支持哪些标准应用程序服务，语法如下所示。大括号内列出每个受支持的服务，每行一个服务，后跟“Initiate”或“Execute”，以说明该服务是否可以启动，执行或启动和执行同时进行。

BACnet Standard Application Services Supported：**↵**

{**↵**

**□** Initiate**↵**

**□** Execute**↵**

**□** Initiate Execute**↵**

}

标准服务可以是第21条BACnetServicesSupported中列出的任何服务。

格式应与标准中每个相应服务部分的标题相匹配，删去文本’Service’。例如，如果设备支持AcknowledgeAlarm服务，则EPICS的此部分应包括文本’AcknowledgeAlarm’。

例如：

BACnet Standard Application Services Supported: **↵** 

{**↵**

Who-Is Initiate Execute**↵**

I-Am Initiate Execute**↵**

Who-Has Execute**↵**

I-Have Initiate**↵** 

ReadProperty Execute**↵**

ReadPropertyMultiple Execute**↵**

WriteProperty Execute**↵**

}

1. 支持的对象类型

本部分说明支持哪些标准对象类型，语法如下所示。大括号内列出每个受支持的对象类型，每行一个对象类型，后跟“Createable”或“Deleteable”，或二者皆可，以说明该对象类型支持动态创建或删除。

Standard Object Types Supported: **↵** 

{**↵**

**□↵**

**□** Createable**↵**

**□** Deleteable**↵**

**□** Createable Deleteable**↵**

}**↵**

标准对象可以是第21条BACnetObjectTypesSupported中列出的任何服务。

格式应与标准中每个相应标准类型部分的标题相匹配，并删去文本“Service”。例如，如果设备支持对象访问方式，则EPICS的此部分应包括文本’Access Door’。

例如：

BACnet Standard Application Services Supported: **↵** 

{**↵**

Analog Value Createable Deleteable**↵**

Analog Input**↵**

Device**↵**

}

1. 数据链路层类型

本部分说明支持哪些标准数据链路层类型，语法如下所示。大括号内列出每个受支持的数据链路层类型，每行一个数据链路层类型。MS/TP和点对点数据链路还需要指定其支持的波特率范围。

Data Link Layer Option: **↵** 

{**↵**

ISO 8802-3, 10BASE5**↵**

ISO 8802-3, 10BASE2**↵**

ISO 8802-3, 10BASET**↵**

ISO 8802-3, fiber**↵**

ARCNET, coax star**↵**

ARCNET, coax bus**↵**

ARCNET, twisted pair star**↵**

ARCNET, twisted pair bus**↵**

ARCNET, fiber star**↵**

ARCNET, twisted pair, EIA-485, Baud rate(s):**□↵** 

MS/TP master. Baud rate(s): 9600,**□↵** 

MS/TP slave. Baud rate(s): 9600,**□↵** 

Point-To-Point. EIA 232, Baud rate(s):**□↵** 

Point-To-Point. Modem, Baud rate(s):**□↵**

Point-To-Point. Modem, Autobaud range:**□**to**□↵** 

BACnet/IP, 'DIX' Ethernet**↵**

BACnet/IP, Other**↵**

Other**↵**

}**↵**

1. 字符集

本部分说明支持哪些BACnet字符集，语法如下所示。大括号内列出每个受支持的字符类型，每行一个。

Character Sets Supported: **↵**

{**↵**

ANSI X3.4**↵**

IBM/Microsoft DBCS**↵**

JIS C 6226**↵**

ISO 8859-1 **↵**

ISO 10646 (UCS-4) **↵**

ISO 10646 (UCS2) **↵**

}**↵**

1. 特殊功能

本部分说明支持哪些BACnet特殊功能集，语法如下所示。大括号内列出每个受支持的特殊功能，每行一个。最大APDU尺寸和窗口尺寸应指定为整数。

Special Functionality: **↵**

{**↵**

Maximum APDU size in octets: **□ ↵**

Segmented Requests Supported， window size: **□ ↵**

Segmented Responses Supported， window size: **□↵**

Router**↵**

BACnet/IP BBMD**↵**

}**↵**

1. 属性值的限制要求

本节定义了对可写属性值的默认限制要求。针对特定数据类型列出限制适用于该数据类型的每个可写属性或可写属性的元素。通过在EPICS的测试数据库部分中添加专门针对该属性的新限制要求，可以覆盖该属性之前的限制要求，见5.4.2节。应该只列出那些定义了默认限制的数据类型，每行一个数据类型。空列表表示无默认限制要求。

Default Property Value Restrictions: **↵**

{**↵**

unsigned-integer: <minimum: **□**; maximum: **□**>**↵**

signed-integer: <minimum: **□**; maximum: **□**>**↵**

real: <minimum: **□**; maximum: **□**; resolution: **□**>**↵**

double: <minimum: **□**; maximum: **□**; resolution: **□**>**↵**

date: <minimum: **□**; maximum: **□**>**↵**

octet-string: <maximum length string: **□**>**↵**

character-string: <maximum length string: **□**>**↵**

list: <maximum length list: **□** >**↵**

variable-length-array: <maximum length array: **□** >**↵**

}**↵**

1. 定时器

本节定义了用于确定由于TD未被观察到正确响应使测试失败的定时器时间。所有定时器值必须是以秒为单位的实数值。参照6.3节。

Fail Times: **↵**

{**↵**

Notification Fail Time: **□↵**

Internal Processing Fail Time: **□↵**

Minimum ON/OFF Time: **□↵**

Schedule Evaluation Fail Time: **□↵**

External Command Fail Time: **□↵**

Program Object State Change Fail Time: **□↵**

Acknowledgement Fail Time: **□↵**

Slave Proxy Confirm Interval: **□↵**

}**↵**

1. 测试数据库

EPICS文件的最后一部分内容定义了设备的测试数据库的对象及其属性。本部分的语法如下。

List of Objects in Test Device: **↵**

{**↵**

*object1***↵**

*object2***↵**

...

*objectN***↵**

}**↵**

每个对象都是由一个包含在大括号内的对象属性值的集合来定义。出现在大括号内的第一个属性必须是Object\_Identifier，它指定元组（object type，instance）。第二个属性应始终为Object\_Name。第三属性始终为Object\_Type，其值应与对象标识符元组的object type部分一致。

{

object-identifier: (*object-type，instance*)

object-name: “**□**”

object-type: object-type

other properties...

}

非标准对象的定义只包含BACnet标准要求的三个属性，如下所示:

{

object-identifier: (proprietary **□**， *instance*)

object-name: “**□**”

object-type: proprietary **□**

}

在测试数据库中，可写的属性应在其属性值后带有一个“W”，如下所示:

{

object-identifier: (analog-value， 6)

object-name: “**□**”

object-type: analog-value

present-value: 23.4 W

other properties...

}

在测试数据库中，有条件可写的属性在其属性值后面必须带有一个“C”，如下所示。建议在注释中标明管理方法。

{

object-identifier: (analog-input， 6)

object-name: “**□**”

object-type: analog-input

present-value: 12.3 C -- Writable when Out\_Of\_Service is TRUE

other properties...

}

以下各节说明每种标准对象类型的模板。为了提高可读性，示例中不再显示回车符/换行符对。

1. 累加器

{

object-identifier: (accumulator， **□**)

object-name: “**□**”

object-type: accumulator

present-value: **□**

description: “**□**”

device-type: “**□**”

status-flags: **□**

event-state: **□**

reliability: **□**

out-of-service: **□**

scale: **□**

units: **□**

prescale: **□**

max-pres-value: **□**

value-change-time: **□**

value-before-change: **□**

value-set: **□**

logging-record: **□**

logging-object: **□**

pulse-rate: **□**

high-limit: **□**

low-limit: **□** 

limit-monitoring-interval: **□** 

notification-class: **□** 

time-delay: **□** 

limit-enable: **□** 

event-enable: {**□**,**□**,**□**}

acked-transitions: {**□**,**□**,**□**}

notify-type: **□**

event-time-stamps: {**□**,**□**,**□**}

profile-name: "**□**"

}

1. 模拟输入

{

object-identifier: (analog-input， **□**)

object-name: “**□**”

object-type: analog-input

present-value: ？

description: “**□**”

device-type: “**□**”

status-flags: {**□**，**□**，**□**，**□**}

event-state: **□**

reliability: **□**

out-of-service: **□**

update-interval: **□**

units: **□**

min-pres-value: **□**

max-pres-value: **□**

resolution: **□**

cov-increment: **□**

time-delay: **□**

notification-class: **□**

high-limit: **□**

low-limit: **□**

deadband: **□**

limit-enable: {**□**，**□**}

event-enable: {**□**，**□**，**□**}

acked-transitions: {**□**，**□**，**□**}

notify-type: **□**

event-time-stamps: {**□**，**□**，**□**}

profile-name: “**□**”

}

1. 模拟输出

{

object-identifier: (analog-output， **□**)

object-name: “**□**”

object-type: analog-output

present-value: **□**

description: “**□**”

device-type: “**□**”

status-flags: {**□**，**□**，**□□**，**□**}

event-state: **□**

reliability: **□**

out-of-service: **□**

units: **□**

min-pres-value: **□**

max-pres-value: **□**

resolution: **□**

priority-array: {**□**，**□**，?，?，**□**，?，?，**□**，?，?，?，?，?，?，?，?}

relinquish-default: **□**

cov-increment: **□**

time-delay: **□**

notification-class: **□**

high-limit: ?

low-limit: **□**

deadband: **□**

limit-enable: {**□**，**□**}

event-enable: {**□**，**□**，**□**}

acked-transitions: {**□**，**□**，**□**}

notify-type: **□**

event-time-stamps: {**□**，**□**，**□**}

profile-name: “**□**”

}

1. 模拟值

{

object-identifier: (analog-value， **□**)

object-name: “**□**”

object-type: analog-value

present-value: **□**

description: “**□**”

status-flags: {**□**，**□**，**□**，**□**}

event-state: **□**

reliability: **□**

out-of-service: **□**

units: **□**

priority-array: {**□**，**□**，？，？，**□**，？，？，**□**，？，？，？，？，？，？，？，？}

relinquish-default: **□**

cov-increment: **□**

time-delay: **□**

notification-class: **□**

high-limit: **□**

low-limit: **□**

deadband: **□**

limit-enable: {**□**，**□**}

event-enable: {**□**，**□**，**□**}

acked-transitions: {**□**，**□**，**□**}

notify-type: **□**

event-time-stamps: {**□**，**□**，**□**}

profile-name: “**□**”

}

1. 平均值

{

object-identifier: (averaging， **□**)

object-name: “**□**”

object-type: averaging

minimum-value: **□**

minimum-value-timestamp: {**□**，**□**}

average-value: **□**

variance-value: **□**

maximum-value: **□**

maximum-value-timestamp: {**□**，**□**}

description: “**□**”

attempted-samples: **□**

valid-samples: **□**

object-property-reference: **□**

window-interval: **□**

window-samples: **□**

profile-name: “**□**”

}

1. 二进制输入

{

object-identifier: (binary-input， **□**)

object-name: “**□**”

object-type: binary-input

present-value: ？

description: “**□**”

device-type: “**□**”

status-flags: {**□**，**□**，**□**，**□**}

event-state: **□**

reliability: **□**

out-of-service: **□**

polarity: **□**

inactive-text: “**□**”

active-text: “**□**”

change-of-state-time: {**□**，**□**}

change-of-state-count: **□**

time-of-state-count-reset: {**□**，**□**}

elapsed-active-time: **□**

time-of-active-time-reset: {**□**，**□**}

time-delay: **□**

notification-class: **□**

alarm-value: **□**

event-enable: {**□**，**□**，**□**}

acked-transitions: {**□**，**□**，**□**}

notify-type: **□**

event-time-stamps: {**□**，**□**，**□**}

profile-name: “**□**”

}

1. 二进制输出

{

object-identifier: (binary-output， **□**)

object-name: “**□**”

object-type: binary-output

present-value: ？

description: “**□**”

device-type: “**□**”

status-flags: {**□**，**□**，**□**，**□**}

event-state: **□**

reliability: **□**

out-of-service: **□**

polarity: **□**

inactive-text: “**□**”

active-text: “**□**”

change-of-state-time: {**□**，**□**}

change-of-state-count: **□**

time-of-state-count-reset: {**□**，**□**}

elapsed-active-time: **□**

time-of-active-time-reset: {**□**，**□**}

minimum-off-time: **□**

minimum-on-time: **□**

priority-array: {**□**，**□**，？，？，**□**，？，？，**□**，？，？，？，？，？，？，？，？}

relinquish-default: **□**

time-delay: **□**

notification-class: **□**

feedback-value: ？

event-enable: {**□**，**□**，**□**}

acked-transitions: {**□**，**□**，**□**}

notify-type: **□**

event-time-stamps: {**□**，**□**，**□**}

profile-name: “**□**”

}

1. 二进制值

{

object-identifier: (binary-value， **□**)

object-name: “**□**”

object-type: binary-value

present-value: ？

description: “**□**”

status-flags: {**□**，**□**，**□**，**□**}

event-state: **□**

reliability: **□**

out-of-service: **□**

inactive-text: “**□**”

active-text: “**□**”

change-of-state-time: {**□**，**□**}

change-of-state-count: **□**

time-of-state-count-reset: {**□**，**□**}

elapsed-active-time: **□**

time-of-active-time-reset: {**□**，**□**}

minimum-off-time: **□**

minimum-on-time: **□**

priority-array: {**□**，**□**，？，？，**□**，？，？，**□**，？，？，？，？，？，？，？，？}

relinquish-default: **□**

time-delay: **□**

notification-class: **□**

alarm-value: **□**

event-enable: {**□**，**□**，**□**}

acked-transitions: {**□**，**□**，**□**}

notify-type: **□**

event-time-stamps: {**□**，**□**，**□**}

profile-name: “**□**”

}

1. 日历

{

object-identifier: (calendar， **□**)

object-name: “**□**”

object-type: calendar

description: “**□**”

present-value: **□**

date-list: (**□**，**□**…)

profile-name: “**□**”

}

1. 命令

{

object-identifier: (command， **□**)

object-name: “**□**”

object-type: command

description: “**□**”

present-value: **□**

in-process: **□**

all-writes-successful: **□**

action: {**□**，**□**…}

action-text: {"**□**","**□**"...}

profile-name: "**□**"

}

1. 设备

{

object-identifier: (device， **□**)

object-name: “**□**”

object-type: device

system-status: **□**

vendor-name: “**□**”

vendor-identifier: **□**

model-name: “**□**”

firmware-revision: “**□**”

application-software-version: “**□**”

location: “**□**”

description: “**□**”

protocol-version: **□**

protocol-revision: **□**

protocol-services-supported: {**□**，**□**...}

protocol-object-types-supported: {**□**，**□**...}

object-list: {**□**， **□**...}

max-APDU-length-accepted: **□**

segmentation-supported: **□**

vt-classes-supported: (**□**， **□**...)

active-vt-sessions: (**□**， **□**...)

local-time: **□**

local-date: **□**

utc-offset: **□**

daylight-savings-status: **□**

apdu-segment-timeout: **□**

apdu-timeout: **□**

number-of-APDU-retries: **□**

list-of-session-keys: (**□**， **□**...)

time-synchronization-recipients: (**□**， **□**...)

max-master: **□**

max-info-frames: **□**

device-address-binding: (**□**， **□**...)

database-revision: **□**

configuration-files: **□**

last-restore-time: **□**

backup-failure-timeout: **□**

active-cov-subscriptions: **□**

profile-name: “**□**”

}

1. 事件注册

{

object-identifier: (event-enrollment， **□**)

object-name: “**□**”

object-type: event-enrollment

description: “**□**”

event-type: **□**

notify-type: **□**

event-parameters: {**□**，**□**...}

object-property-reference: (**□**)

event-state: **□**

event-enable: {**□**，**□**，**□**}

acked-transitions: {**□**，**□**，**□**}

notification-class: **□**

-- The following four properties were removed from Event Enrollment objects in protocol revision 4:

-- recipient: **□**

-- process-identifier: **□**

-- priority: **□**

-- issue-confirmed-notifications: **□**

event-time-stamps: {**□**，**□**，**□**}

profile-name: “**□**”

}

1. 文件

{

object-identifier: (file， **□**)

object-name: “**□**”

object-type: file

description: “**□**”

file-type: “**□**”

file-size: **□**

modification-date: {**□**，**□**}

archive: **□**

read-only: **□**

file-access-method: **□**

profile-name: “**□**”

}

1. 组

{

object-identifier: (group， **□**)

object-name: “**□**”

object-type: group

description: “**□**”

list-of-group-members: (**□**， **□**...)

present-value: (**□**， **□**...)

profile-name: “**□**”

}

1. 生命安全点

{

object-identifier: (life-safety-point， **□**)

object-name: “**□**”

object-type: life-safety-point

present-value: ？

tracking-value: **□**

description: “**□**”

device-type: “**□**”

status-flags: {**□**，**□**，**□**，**□**}

event-state: **□**

reliability: **□**

out-of-service: **□**

mode: **□**

time-delay: **□**

notification-class: **□**

life-safety-alarm-values: (**□**，**□**...)

alarm-values: (**□**，**□**...)

fault-values: (**□**，**□**...)

event-enable: {**□**，**□**，**□**}

acked-transitions: {**□**，**□**，**□**}

notify-type: **□**

event-time-stamps: {**□**，**□**，**□**}

silenced: **□**

operation-expected: **□**

maintenance-required: **□**

setting: **□**

direct-reading**□**

units: **□**

member-of: (**□**，**□**...)

profile-name: “**□**”

}

1. 生命安全区

{

object-identifier: (life-safety-zone， **□**)

object-name: “**□**”

object-type: life-safety-zone

present-value: ？

tracking-value: **□**

description: “**□**”

device-type: “**□**”

status-flags: {**□**，**□**，**□**，**□**}

event-state: **□**

reliability: **□**

out-of-service: **□**

mode: **□**

time-delay: **□**

notification-class: **□**

life-safety-alarm-values: (**□**，**□**...)

alarm-values: (**□**，**□**...)

fault-values: (**□**，**□**...)

event-enable: (**□**，**□**，**□**)

acked-transitions: (**□**，**□**，**□**)

notify-type: **□**

event-time-stamps: {**□**，**□**，**□**}

silenced: **□**

operation-expected: **□**

maintenance-required: **□**

zone-members: (**□**，**□**...)

member-of: (**□**，**□**...)

profile-name: “**□**”

}

1. 循环

{

object-identifier: (loop， **□**)

object-name: “**□**”

object-type: loop

present-value: ？

description: “**□**”

status-flags: {**□**，**□**，**□**，**□**}

event-state: **□**

reliability: **□**

out-of-service: **□**

update-interval: **□**

output-units: **□**

manipulated-variable-reference: {**□**}

controlled-variable-reference: {**□**}

controlled-variable-value: **□**

controlled-variable-units: **□**

setpoint-reference: {**□**}

setpoint: **□**

action: **□**

proportional-constant: **□**

proportional-constant-units: **□**

integral-constant: **□**

integral-constant-units: **□**

derivative-constant: **□**

derivative-constant-units: **□**

bias: **□**

maximum-output: **□**

minimum-output: **□**

priority-for-writing: **□**

cov-increment: **□**

time-delay: **□**

notification-class: **□**

error-limit: **□**

event-enable: {**□**，**□**，**□**}

acked-transitions: {**□**，**□**，**□**}

notify-type: **□**

event-time-stamps: {**□**，**□**，**□**}

profile-name: “**□**”

}

1. 多状态输入

{

object-identifier: (multi-state-input， **□**)

object-name: “**□**”

object-type: multi-state-input

present-value: ？

description: “**□**”

device-type: “**□**”

status-flags: {**□**，**□**，**□**，**□**}

event-state: **□**

reliability: **□**

out-of-service: **□**

number-of-states: **□**

state-text: {“**□**”， “**□**”...}

time-delay: **□**

notification-class: **□**

alarm-values: (**□**，**□**…)

fault-values: (**□**，**□**…)

event-enable: {**□**，**□**，**□**}

acked-transitions: {**□**，**□**，**□**}

notify-type: **□**

event-time-stamps: {**□**，**□**，**□**}

profile-name: “**□**”

}

1. 多状态输出

{

object-identifier: (multi-state-output， **□**)

object-name: “**□**”

object-type: multi-state-output

present-value: ？

description: “**□**”

device-type: “**□**”

status-flags: {**□**，**□**，**□**，**□**}

event-state: **□**

reliability: **□**

out-of-service: **□**

number-of-states: **□**

state-text: {“**□**”， “**□**”...}

priority-array: {**□**，**□**，？，？，**□**，？，？，**□**，？，？，？，？，？，？，？，？}

relinquish-default: **□**

time-delay: **□**

notification-class: **□**

feedback-value: **□**

event-enable: {**□**，**□**，**□**}

acked-transitions: {**□**，**□**，**□**}

notify-type: **□**

event-time-stamps: {**□**，**□**，**□**}

profile-name: “**□**”

}

1. 多状态值

{

object-identifier: (multi-state-value， **□**)

object-name: “**□**”

object-type: multi-state-value

present-value: ？

description: “**□**”

device-type: “**□**”

status-flags: {**□**，**□**，**□**，**□**}

event-state: **□**

reliability: **□**

out-of-service: **□**

number-of-states: **□**

state-text: {“**□**”， “**□**”...}

priority-array: {**□**，**□**，？，？，**□**，？，？，**□**，？，？，？，？，？，？，？，？}

relinquish-default: **□**

time-delay: **□**

notification-class: **□**

alarm-values: (**□**，**□**...)

fault-values: (**□**，**□**...)

feedback-value: **□**

event-enable: {**□**，**□**，**□**}

acked-transitions: {**□**，**□**，**□**}

notify-type: **□**

event-time-stamps: {**□**，**□**，**□**}

profile-name: “**□**”

}

1. 通知类

{

object-identifier: (notification-class， **□**)

object-name: “**□**”

object-type: notification-class

description: “**□**”

notification-class: **□**

priority: {**□**， **□**， **□**}

ack-required: {**□**，**□**，**□**}

recipient-list: (**□**，**□**…)

profile-name: “**□**”

}

1. 程序

{

object-identifier: (program， **□**)

object-name: “**□**”

object-type: program

program-state: **□**

program-change; **□**

reason-for-halt: **□**

description-of-halt: “**□**”

program-location: “**□**”

description: “**□**”

instance-of: “**□**”

status-flags: {**□**，**□**，**□**，**□**}

reliability: **□**

out-of-service: **□**

profile-name: “**□**”}

1. 脉冲转换器

{

object-identifier: (pulse-converter， **□**)

object-name: “**□**”

object-type: pulse-converter

description: “**□**”

present-value: **□**

input-reference: **□**

status-flags: **□**

event-state: **□**

reliability: **□**

out-of-service: **□**

units: **□**

scale-factor: **□**

adjust-value: **□**

count: **□**

update-time: **□**

count-change-time: **□**

count-before-change: **□**

cov-increment: **□**

cov-period: **□**

notification-class: **□**

time-delay: **□**

high-limit: **□**

low-limit: **□**

deadband: **□**

limit-enable: **□**

event-enable: {**□**，**□**，**□**}

acked-transitions: {**□**，**□**，**□**}

notify-type: **□**

event-time-stamps: {**□**，**□**，**□**}

profile-name: “**□**”

}

1. 时间表

{

object-identifier: (schedule， **□**)

object-name: “**□**”

object-type: schedule

present-value: **□**

description: “**□**”

effective-period: {**□**，**□**}

weekly-schedule: {**□**，**□**...}

exception-schedule: {**□**，**□**...}

list-of-object-property-references: (**□**，**□**…)

priority-for-writing: **□**

profile-name: “**□**”

}

1. 趋势日志

{

object-identifier: (trend-log， **□**)

object-name: “**□**”

object-type: trend-log

description: “**□**”

log-enable: **□**

start-time: **□**

stop-time: **□**

log-device-object-property: {**□**}

log-interval: **□**

cov-resubscription-interval : **□**

client-cov-increment: **□**

stop-when-full: **□**

buffer-size: **□**

log-buffer: **□**

record-count: **□**

total-record-count: **□**

notification-threshold: **□**

records-since-notification: **□**

previous-notify-time: **□**

current-notify-time: **□**

event-state: **□**

notification-class: **□**

event-enable: {**□**，**□**，**□**}

acked-transitions: {**□**，**□**，**□**}

notify-type: **□**

event-time-stamps: {**□**，**□**，**□**}

profile-name: “**□**”

}

1. 访问口

{

object-identifier: (access-door， **□**)

object-name: “**□**”

object-type: access-door

present-value : **□**

description: “**□**”

status-flags: **□**

event-state: **□**

reliability: **□**

out-of-service: **□**

priority-array: **□**

relinquish-default: **□**

door-status: **□**

lock-status: **□**

secured-status: **□**

door-members: {**□**， **□**...}

door-pulse-time: **□**

door-extended-pulse-time: **□**

door-unlock-delay-time: **□**

door-open-too-long-time: **□**

door-alarm-state: **□**

masked-alarm-values: {**□**， **□** ...}

maintenance-required: **□**

time-delay: **□**

notification-class: **□**

alarm-values: {**□**， **□**...}

fault-values: {**□**， **□**...}

event-enable: {**□**，**□**，**□**}

acked-transitions: {**□**，**□**，**□**}

notify-type: **□**

event-time-stamps: {**□**，**□**，**□**}

profile-name: “**□**”

}

1. 负载控制

{

object-identifier: (load-control， **□**)

object-name: “**□**”

object-type: load-control

description: “**□**”

present-value: **□**

state-description: “**□**”

status-flags: **□**

event-state: **□**

reliability: **□**

requested-shed-level: **□**

start-time: **□**

shed-duration: **□**

duty-window: **□**

enabled: **□**

full-duty-baseline: **□**

expected-shed-level: **□**

actual-shed-level: **□**

shed-levels: {**□**， **□**...}

shed-level-descriptions: {“**□**”， “**□**”...}

notification-class: **□**

time-delay: **□**

event-enable: {**□**，**□**，**□**}

acked-transitions: {**□**，**□**，**□**}

notify-type: **□**

event-time-stamps: {**□**，**□**，**□**}

profile-name: “**□**”

}

1. 结构化视图

{

object-identifier: (structured-view， **□**)

object-name: “**□**”

object-type: structured-view

description: “**□**”

node-type: **□**

node-subtype: “**□**”

subordinate-list: {**□**， **□**...}

subordinate-annotations: {“**□**”， “**□**”...}

profile-name: “**□**”

}

1. 协议实现一致性声明EPICS的测试内容

设备的每个执行操作都应进行测试，以确保相关数据元素之间的一致性。这些测试包括：

（a）特定的BIBBs所需的所有对象类型是否全部都在EPICS标准对象类型支持的部分中。

（b）测试数据库应至少包括指定BIBBs需要的每种对象类型的一个实例。

（c）测试数据库中设备对象的Object\_Types\_Supported属性应说明对受支持的BIBBs所需的每种对象类型的支持。

（d）在EPICS的BACnet Standard Application Services Supported部分中，应说明受支持的BIBBs所需的所有应用服务，并按照受支持的BIBBs的需要注明“Initiate”和“Execute”。

（e）测试数据库中Device对象的Application\_Services\_Supported属性应说明对受支持的BIBBs所需执行的应用程序服务的支持。

（f）EPICS的“Standard Object Types Supported”部分中列出的对象类型应与测试数据库中包含的设备对象的“Object\_Types\_Supported”属性中列出的对象类型一一对应。

（g）对于EPICS的“Standard Object Types Supported”部分中列出的每种对象类型，测试数据库中应至少具有该类型的一个对象。

（h）设备对象的Object\_List属性中列出的对象与测试数据库中包含的对象应一一对应。 Object\_List属性和测试数据库都应包括所有的专有对象。BACnet条款23.4.3中不必需的专有对象的属性不需要包含在测试数据库中。

（i）对于测试数据库中包含的每个对象，应提供BACnet第12条中定义的该对象的所有必需属性。此外，如果对象支持的任何属性需要其他属性的条件存在，则应验证它们的存在。

（j）对于每个要求可写的属性，该属性应在EPICS中标记为可写。

（k）Protocol\_Services\_Supported位串的长度为IUT声明的协议修订版的BACnetProtocolServicesSupported定义的位数。

（l）Protocol\_Object\_Types\_Supported位串的长度为IUT声明的协议修订版的BACnetObjectTypesSupported定义的位数。

1. 一致性测试脚本语言TCSL的规定

为简化和阐明测试说明，规定一个简单的一致性测试脚本语言（TCSL）。按照编程语言语法的巴科斯范式格式（BNF），本脚本语言使用以下符号：

<part> 语言组件名称用尖括号括起来

::= 被定义为

<a> <b> <c> 隐式连接

| 模式选择

( ) 必填部分

[ ] 选填部分

( )… 至少填一个，可重复

[ ]… 选填，可重复

<d>… 元素出现一次，可重复

'?' 符号用单引号括起来

WOW 保留字大写

-- 表示此行其余字符是注释

因为TCSL是伪语言，而非TD的实现语言，所以巴科斯范式格式可以不那么严格，某些部分可使用英文短语或句子。

在使用TCSL描述的测试中，除非有明确的附加约束，否则完全按规定交换消息构成传递结果。

* 1. 测试脚本语言TCSL内容

1. 常用的符号和字符

以下内容定义了常用符号和字符。

<binary digit> ::= '0' | '1'

<decimal digit> ::= '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'

<hex digit> ::= <decimal digit> | 'A' | 'B' | 'C' | 'D' | 'E' | 'F'

<single quote> ::= (the single quote character)

<double quote> ::= (the double quote character)

<conditional> ::= '=' | '<' | '>' | '<=' | '>=' | '~=' | '<>'

条件“〜=”为约等于，其近似定义取决于上下文。

1. 整数

以下内容定义了整数，这些定义与BACnet标准中使用的语法一致。

<integer> ::= <binary int>|<decimal int>|<hex int>

<binary int> ::= B<single quote> <binary digit>…<single quote>

<decimal int> ::= [’-’]<unsigned>|D<single quote><decimal digit>…<single quote>

<hex int> ::= X<single quote><hex digit>…<single quote>

<unsigned> ::= <decimal digit>…

1. 文本字符串

表示属性值和服务参数值的文本字符串可以是大写字母、小写字母或大小写的混合，且要用双引号括起来。

例如：Object\_Name = "CW\_STEMP", Description = "AC1 Supply Temperature"。

1. 枚举

枚举类型的属性值和参数值必须用大写字母表示，不需要双引号。

例如: TRUE，FALSE，RELIABLE，UNRELIABLE。

1. 属性标识符

属性标识符由大小写字母混合表示，单词首字母大写，若与其他单词共同构成一个属性标识符，则单词之间要用下划线（\_）隔开。

例如：Present\_Value，Reliability，Object\_Identifier，Object\_Type，和Vendor\_Name。

1. 服务参数

服务参数名要用单引号括起来。

例如:’Return Read Access Specifications with Result’，’List of Read Access Results’。

1. 对象标识符

对象标识符的表示定义如下:

<object identifier> ::= '(' <object type> ',' <instance number> ')'

其中

<object type> ::= (one of the object types defined in BACnet Clause 12)

<instance number> ::= <unsigned>

例如:(Analog Input,1)，(Device, 150)

* 1. 测试脚本语言TCSL语句

TCSL中的语句分为两类，一类是控制测试流程和顺序，另一类通知TD发送/接收数据。

<statement> ::= <simple statement> | <compound statement>

<compound statement> ::= '{' <statement>… '}'

<compound statement>可以用在任何可以使用<simple statement>的地方。

<simple statement> ::= <if statement> | <repeat statement> | <error statement>

| <check statement> | <make statement>

| <transmit statement> | <receive statement>

| <write statement> | <verify statement> | <before statement>

| <read statement>

1. IF语句

IF语句用于测试某个条件，并根据测试结果选择继续向下的控制流程。

<if statement> ::= IF ’(’ <condition> ’)’ THEN <statement> [ ELSE <statement> ]

<condition>为TD需要做出的选择。如果<condition>为true，则执行THEN后面的语句。如果<condition>为false且已指定ELSE子句，则执行ELSE语句后的语句，例如：

IF (it is raining outside) THEN

VERIFY Present\_Value = 1

ELSE

VERIFY Present\_Value = 2

1. REPEAT语句

REPEAT语句用于循环访问相同的对象或数值。

<repeat statement> ::= REPEAT <var> ’=’ ’(’ <list description> ’)’ DO <statement>

<var>是特殊标识符，必须为大写字母，通常是字母“X”、“Y”或“Z”，可取<list description>中的每个值，<list description>表示迭代元素。例如：

REPEAT X = (values specified by 6.3 appropriate to the object type) DO {

WRITE Present\_Value=X

VERIFY Present\_Value=X

}

1. ERROR语句

若TCSL步骤产生错误情况，该错误情况可以通过ERROR语句查看。

<error statement> ::= ERROR [ <explanation string> ]

<explanation string>是提供给TD管理员进行错误诊断的文本字符串，可选。例如：

ERROR “Retry count exceeded”

1. CHECK语句

当管理员要验证TD的某些操作导致了网络不可见的IUT更改时，要使用CHECK语句。

<check statement> ::= CHECK ’(’ <condition> ’)’

管理员可以通知TD操作是否成功，<condition>描述要检查的内容。如果操作不成功，则测试失败。例如：

CHECK (Did the IUT reboot?)

1. MAKE语句

当管理员要在IUT中进行创建或更改操作时，要使用MAKE语句。

<make statement> ::= MAKE ’(’ <action> ’)’

<action>即为要进行的操作，例如：

MAKE (Out\_Of\_Service TRUE)

1. TRANSMIT语句

TRANSMIT语句用于发送数据包。

<transmit statement> ::= TRANSMIT <packet desc>

其中

<packet desc> ::= [ <port> ’,’] [ <addressing> ’,’] ( <service specification> | <pdu specification> | <string>)

<port> ::= PORT <port identifier>

<port identifier> ::= ’A’ | ’B’ ... ’Z’

<addressing> ::= ( <dst> | <src> | <dst> ’,’ <src> )

<src> ::= SOURCE ’=’ <src parm value>

<src parm value> ::= TD|IUT

<dst> ::= DESTINATION ’=’ <dst parm value>

<dst parm value> ::= LOCAL BROADCAST|GLOBAL BROADCAST

|REMOTE BROADCAST <net>

|IUT|TD

<net> ::= (a valid BACnet network number)

<service specification> ::= <BACnet service> [ ’,’ <pdu parm list> ] [ ’,’ <service parm list> ]

<BACnet service> ::= (any BACnet service choice)

<service parm list> ::= <service parameter> ’=’ <parameter value> [’,’ <service parm list> ]

<service parameter> ::= (parameter name specific to the BACnet service)

<pdu specification> ::= <pdu type> [ ’,’ <pdu parm list> ]

<pdu type> ::= (any BACnet application，network，link，or MAC layer PDU type)

<pdu parm list> ::= <pdu parameter> ’=’ <parameter value> [’,’ <pdu parm list> ]

<pdu parameter> ::= (any BACnet application, network, data link, or MAC layer PDU parameter)

<parameter value> ::= ( <atomic value> | <parameter value list> | <parameter cond value> )

<parameter value list> ::= ’(’ <parameter value> [ ’|’ <parameter value> ]... ’)’

<parameter cond value> ::= ’(’ IF <condition> THEN <parameter value> ELSE <parameter value> ’)’

<string> ::= <”> <ASCII Char> <”>

<ASCII Char> ::= (ANSI X3.4 character)

SOURCE和DESTINATION参数用于指定NPDU，LPDU和MPDU参数值的常见组合。如果未指定DESTINATION和SOURCE，则源地址为TD，目标地址为IUT。

<pdu parameter>=<parameter value>列表表示用指定参数传送指定值，参数值可用任意顺序指定。

<service parameter>=<parameter value>列表表示用指定参数传送指定值，参数值可用任意顺序指定。

例1:

TRANSMIT DESTINATION = GLOBAL BROADCAST, Who-Is

在这个简单的例子中，Who-Is服务没有强制性参数，根据定义<pdu type>是为BACnet-Unconfirmed-Request-PDU。DESTINATION表示NPDU，LPDU和MPDU层上的参数值。下面的语句与上述功能相同，但其指定内容更加完整。

TRANSMIT

DA = LOCAL BROADCAST，

SA = TD，

DNET = GLOBAL BROADCAST，

BACnet-Unconfirmed-Request-PDU，

’Service Choice’ = Who-Is

例2:

TRANSMIT ReadProperty-Request,

’Object Identifier’ = (Analog Input,1),

’Property Identifier’ = Present\_Value

在此例中，ReadProperty服务请求使用指定的服务参数值从TD发送到IUT。

1. RECEIVE语句

RECEIVE语句用于定义来自IUT的消息。

<receive statement> ::= RECEIVE ( <packet desc> |’(’ <packet desc> ’)’ [ ’|’ ’(’ <packet desc> ’)’ ] ...)

<pdu specification> 参数与TRANSMIT语句中使用的参数相同。如果未指定，则SOURCE默认为IUT，DESTINATION默认为TD。

例如: TRANSMIT SubscribeCOV-Request,

‘Subscriber Process Identifier’ = any value selected by the TD,

‘Monitored Object Identifier’ = any object supporting COV notification,

‘Issue Confirmed Notifications’ = TRUE,

‘Lifetime’= 0

RECEIVE ConfirmedCOVNotification-Request,

‘Subscriber Process Identifier’ = the value from the previous subscription,

‘Monitored Object’ = the value from the previous subscription,

‘Initiating Device Identifier’ = IUT,

‘Lifetime’ = 0,

‘List of Values’ = values appropriate to the object type of the

monitored object

1. WAIT语句

WAIT语句用于在指定的一段时间内暂停执行TD。

<wait statement> ::= WAIT <timer value>

测试步骤：在进行下一个测试步骤之前，TD暂停<timer value>参数指定的时间量。<timer value>应为本标准6.3节规定的，或ANSI/ASHRAE 135-2004标准或其他指定标准中规定的时间值。

例如: WAIT **Internal Processing Fail Time**

1. WRITE语句

WRITE语句用于修改一个对象的特定属性值。

<write statement> ::= WRITE [ <object identifier> ’,’ ] <property identifier> ’=’ <property value>

[ ’,’ARRAY INDEX ’=’ <index value> ]

[ ’,’ PRIORITY ’=’ <write priority> ]

这是以下语句的快捷表达：

1. TRANSMIT WriteProperty-Request，

’Object Identifier’ = <object identifier>,

’Property Identifier’ = <property identifier>,

’Property Value’ = <property value>,

’Array Index’ = <array index>,

’Priority’ = <write priority>

2. RECEIVE BACnet-SimpleACK-PDU

注意：在某些测试中，从描述中省略了<object identifier>，因为从上下文中可以清楚地知道应写入哪种对象类型，并且可以接受该对象类型的任何实例。但在不能判断的情况下，应明确指定此参数。

如果<index value>或<write priority>被省略，那么在WriteProperty服务请求里也应省略相应的服务参数。

例如: WRITE (Analog Output，1)，Present\_Value=6.5，PRIORITY=8

1. VERIFY语句

<verify statement> ::= VERIFY [ <object identifier> ’,’ ] <property identifier> <conditional> <property

value> [ ’,’ ARRAY INDEX ’=’ <array index> ]

验证过程步骤如下：

1. WAIT **Internal Processing Fail Time**

2. TRANSMIT ReadProperty-Request,

'Object Identifier' = <object identifier>,

'Property Identifier' = <property identifier>,

'Property Array Index' = <array index>

3. RECEIVE BACnet-ComplexACK-PDU,

'Object Identifier' = <object identifier>,

'Property Identifier' = <property identifier>,

'Property Array Index' = <array index>,

'Property Value' = (any valid value x, where x <conditional> <property value> is TRUE subject to the resolution constraints of 5.4.2)

例如：WRITE (Analog Output, 1), Present\_Value = 6.5, PRIORITY = 8

VERIFY (Analog Output, 1), Present\_Value = 6.5

1. BEFORE语句

BEFORE语句用于在一个定时器设定时间过期之前，测试预期动作是否发生。

<before statement> ::= BEFORE <timer> <statement>

<timer value>应是本标准6.3规定的，或ANSI / ASHRAE 135-1995标准的或其他指定标准中规定的一个定时时间。若超出该时间且<statement>中的操作尚未发生，则测试失败。否则，测试通过，并继续进行测试。例如：

BEFORE **Acknowledgment Fail Time** RECEIVE BACnet-Simple-ACK

1. WHILE语句

WHILE语句用于反复执行一个步骤或一系列的步骤，直到某条件变为FALSE。

<while statement> ::= WHILE ’(’ <condition> ’)’ DO <statement>

例如:

WHILE (IUT not initialized) DO {

TRANSMIT Poll For Master

}

1. READ语句

<read statement> ::= READ <variable> '=' [ <object identifier> ',' ] <property identifier>

[ ',' ARRAY INDEX '=' <array index>]

读取过程步骤如下：

1. WAIT **Internal Processing Fail Time**

2. TRANSMIT ReadProperty-Request,

'Object Identifier' = <object identifier>,

'Property Identifier' = <property identifier>,

'Property Array Index' = <array index>

3. RECEIVE BACnet-ComplexACK-PDU,

'Object Identifier' = <object identifier>,

'Property Identifier' = <property identifier>,

'Property Array Index' = <array index>,

'Property Value' = (any valid value, <variable>)

例如：READ X = Priority\_Array, ARRAY\_INDEX=10

* 1. 所需时间测试

BACnet标准没有定义由服务请求操作导致的操作变为网络可见状态应该花多长时间。我们合理地推测这些时间延迟随请求操作实现的不同而不同。在测试环境中，有必要给时间延迟设置一个界限，来判别该测试是否通过。本节定义定时器的目的于此。供应商应为EPICS中的特定功能的实现提供定时器的值，参见5.5.9。这些定时器的数据类型为实值。

1. Notification Fail Time通知失败时间

通知失败时间是指，从观察到事件发生或值改变到因未按预期发送通知而使测试失败之间的时间，（以秒为单位）。

1. Internal Processing Fail Time内部处理失败时间

内部处理失败时间是从收到写入BACnet属性的指令或其他事件更改属性值，到因属性值未更新而导致测试失败的时间（以秒为单位）。

1. Minimum ON/OFF Fail Time最小开/关失败时间

最小开/关失败时间是指，从最小开启或最小关闭定时器时间到期到由于命令优先级6的值不符合预期导致测试失败之间的过程的最大过程时间（以秒为单位）。

1. Schedule Evaluation Fail Time进度评价失败时间

进度评估失败时间是指，从定义日历或时间表的属性或设备的Local\_Time改变，到日历或时间表的Present\_Value不能做出正确响应而使测试失败之间经过的时间（以秒为单位）。

1. External Command Fail Time外部命令失败时间

外部命令失败时间是指，从命令对象的Present\_Value发生改变，到当与新命令状态关联的第一条消息未发送而使测试失败之间经过的时间（以秒为单位）。

1. Program Object State Change Fail Time程序对象状态更改失败时间

程序对象状态更改失败时间是指，从写入Program\_Change属性，到未观察到预期响应而认为测试失败的时间（以秒为单位）。

1. Acknowledgment Fail Time确认失败时间

确认失败时间是指，从发送BACnet确认服务请求，到收到相应的正确响应之间的时间（以秒为单位）。

1. Default Time Delay in Test Descriptions测试描述中的默认延时

对于此标准中定义的测试用例，除非实际规定了时序约束，否则在实际接收到由RECEIVE语句指定的消息之前，可以具有不超过内部处理失败时间的时间延迟。

1. Unconfirmed Response Fail Time未确认响应失败时间

未确认响应失败时间是指，从IUT收到需确认的请求到未接收到来自IUT的确认响应认为测试失败的时间（单位为秒）。例如，IUT发送I-Am请求来响应Who-Is请求所需要的最长时间。

* 1. 测试对应的协议版本

除非特别说明，否则本标准中对BACnet条款的所有引用均参考ANSI/ASHRAE 135-2008（注：“BACnet-2001”指ANSI / ASHRAE 135-2001）。

* 1. TD要求

本标准测试定义了TD和IUT间的特定会话。在进行会话时，IUT可以发起与当前进行的测试相关或者不相关的其他会话。对于由IUT发起的且未在测试步骤中表现的会话，除非在测试中特殊说明，否则TD应生成与其当前模拟的设备能力和状态一致的响应。

1. 一致性测试所涉及的对象属性

应对IUT进行测试，以保证可支持测试数据库的所有对象的属性。测试可以通过读取属性，验证返回值的正确性来判断是否支持；也可以通过写入属性来验证响应是否正确来判断。8.1节介绍了读支持测试方法，8.2节介绍了写支持测试方法。某些BACnet对象也需要根据其属性提供特殊功能，验证此功能的测试将在8.3节中介绍。

* 1. 测试数据库中的读取属性

1. 读支持测试程序

所需测试：ReadProperty服务执行测试，10.18。

目的：验证所有对象的属性可以使用BACnet ReadProperty和ReadPropertyMultiple服务读取。本测试一次采用BACnet ReadProperty执行，一次用BACnet ReadPropertyMultiple执行。在验证数组属性时，应尽可能不使用数组指针。

测试步骤：

1. REPEAT X=(all objects in the IUT's database) DO {

REPEAT Y=(all properties in object X) DO {

VERIFY (X)，Y=(the value for this property specified in the EPICS)

}

}

注释：对于EPICS指出未使用符号“？”指定属性值的情况，则所有的正确数据类型的任何值均应视为匹配。

1. 未注册属性测试

目的：验证EPICS中是否注册了所有对象的全部属性。

测试：对EPICS数据库中的所有对象，测试读取EPICS中没有注册为对象的部分的全部标准属性。

测试步骤：

1. REPEAT X = (a tester selected set of objects) DO {

REPEAT Y = (0 through 511) DO {

IF (the property Y is not in the EPICS for object X) THEN

TRANSMIT ReadProperty-Request,

'Object Identifier' = X,

'Property Identifier' = Y

RECEIVE BACnet-Error-PDU,

Error Class = PROPERTY,

Error Code = UNKNOWN\_PROPERTY

}

}

注释：测试对象应该包括所有支持对象类型的实例。若某对象类型的几个实例支持的属性、属性的允许值范围或属性可写性不同，则选择该对象类型的每个变体的一个实例。

* 1. 测试数据库中的写入属性

1. 属性值功能范围要求

对于每一个可写的属性，测试时将根据本节定义选择多个值来验证其支持的值的范围。

1. 枚举和布尔值

对于枚举和布尔值，应考虑5.4.2中规定的所有允许范围后，对每个枚举值进行测试。

1. 无符号整数，有符号整数，实数，和双精度值

一个连续的数据属性应在上限、下限中选择的两个中间点测试。供应商应为EPICS中的特定服务提供规定数值范围，参见5.4.2。

1. 字节串和字符串

具有八位字节串或字符串数据类型的属性应分别使用一个长度为零的字符串、一个长度为最大支持长度的字符串以及介于两者之间的某个长度的字符串进行测试。供应商应在EPICS中提供最大长度字符串的值，参见5.4.2。

1. 位串

位串数据类型的属性应使用不同于当前属性值的一个值来测试。

1. 日期

日期数据类型的属性应使用不同于当前属性值的一个值来测试。

1. 时间

时间数据类型属性的测试通过写入一个指定字段的时间值来实现。这个写入的时间值应与先前的值相差一个大于IUT时钟分辨率的量。当属性被读回时，时间值应与IUT时钟分辨率范围内的写入值相匹配。

未指定字段的时间值也应被测试。未指定字段应为IUT时钟分辨率范围内的值。对于设备对象的Local\_Time属性以外的所有属性，当回读该值时，时间值应与IUT时钟分辨率范围内的写入值匹配，且未指定的字段保持未指定状态。对与系统时钟耦合的Local\_Time属性的特殊情况，未指定部分的时间返回值是本地问题。

1. 构造数据类型

对于构造的数据类型，测试时应选择要写入一个还是多个值，且写入的值应与数据类型一致。

1. 写支持测试程序

目的：验证可以使用BACnet WriteProperty和WritePropertyMulitiple服务写入任何对象的可写属性。本测试一次采用BACnet ReadProperty执行，一次用BACnet ReadPropertyMultiple执行。在验证数组属性时，应尽可能不使用数组指针。

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

测试步骤：

1. REPEAT X = (all objects in the IUT's database) DO {

REPEAT Y = (all writable properties in object X) DO {

REPEAT Z = (all values meeting the functional range requirements of 7.2.1) DO {

WRITE (X)，Y = Z，

VERIFY (X)，Y = Z

}

}

}

1. 只读属性测试

目的：验证EPICS中标记为只读的属性的确只读。

测试思想：对于EPICS中的每个只读属性（不可写、非条件可写），写入从设备读取的值，验证是否返回错误响应。再写入另一个数据类型允许范围内的值，验证是否返回错误响应。如果属性是列表且IUT支持AddListElemet，则用AddListElemet修改属性并验证是否返回错误响应。若IUT不支持WriteProperty服务则跳过此测试。

测试步骤：

1. REPEAT X = (a tester selected set of objects) DO {

REPEAT Y = (all read-only properties in object X) DO {

IF (the property is not an array) THEN

READ Z = X

TRANSMIT WriteProperty-Request,

'Object Identifier' = X,

'Property Identifier' = Y,

'Property Value' = Z

RECEIVE BACnet-Error-PDU,

Error Class = PROPERTY,

Error Code = WRITE\_ACCESS\_DENIED

TRANSMIT WriteProperty-Request,

'Object Identifier' = X,

'Property Identifier' = Y,

'Property Value' = (any value meeting the range requirements of

7.2.1 except Z)

RECEIVE BACnet-Error-PDU,

Error Class = PROPERTY,

Error Code = WRITE\_ACCESS\_DENIED

IF (the IUT supports AddListElement and the property is a list) THEN

TRANSMIT AddListElement-Request,

'Object Identifier'= X,

'Property Identifier'= Y,

List of Elements' = (any element value meeting the range requirements of 7.2.1 excluding those in Z)

RECEIVE BACnet-Error-PDU,

Error Class = PROPERTY,

Error Code = WRITE\_ACCESS\_DENIED

ELSE

READ LEN = X, Array\_Index = 0

IF (LEN > 0) THEN

READ Z = X, Array Index = 1

TRANSMIT WriteProperty-Request,

'Object Identifier' = X,

'Property Identifier' = Y,

'Property Value' = Z

‘Array Index’ = 1

RECEIVE BACnet-Error-PDU,

Error Class = PROPERTY,

Error Code = WRITE\_ACCESS\_DENIED

TRANSMIT WriteProperty-Request,

'Object Identifier' = X,

'Property Identifier' = Y,

'Property Value' = (any value meeting the range requirements of 7.2.1 except Z)

‘Array Index’ = 1

RECEIVE BACnet-Error-PDU,

Error Class = PROPERTY,

Error Code = WRITE\_ACCESS\_DENIED

IF (the IUT supports AddListElement and the property is an array of lists) THEN

TRANSMIT AddListElement-Request,

'Object Identifier' = X,

'Property Identifier' = Y,

'Array Index' = 1

List of Elements' = (any elements value meeting the range requirements of 7.2.1 excluding those in Z)

RECEIVE BACnet-Error-PDU,

Error Class = PROPERTY,

Error Code = WRITE\_ACCESS\_DENIED

ELSE

TRANSMIT WriteProperty-Request,

'Object Identifier' = X,

'Property Identifier' = Y,

'Property Value' = (any value meeting the range requirements of 7.2.1)

RECEIVE BACnet-Error-PDU,

Error Class = PROPERTY,

Error Code = WRITE\_ACCESS\_DENIED

}

}

注释：测试对象应包括所有支持对象类型的实例，若某对象类型的几个实例支持的属性、属性的允许值范围或属性可写性不同，则选择该对象类型的每个变体的一个实例。

注释：修改属性后，应返回一个错误类型为PROPERTY的错误，错误代码为WRITE\_ACCESS\_DENIED。但IUT可能会返回一个PROPERTY的错误类型替代上述类型，错误代码为VALUE\_OUT\_OF\_RANGE；或者返回一个RESOURCES的错误类型，错误代码为NO\_SPACE\_TO\_WRITE\_PROPERTY。

* 1. 对象功能测试

本节中定义的测试用于验证是否支持各种BACnet对象所需的功能。测试特定于对象类型，在某些情况下还取决于特定属性的值。对于每种支持的对象类型，本节中所有适用的测试都应被执行。测试每种对象类型的单个实例的正确功能即可。

1. 性能测试

本节测试适用于多个对象类型的属性。对于每个支持该属性的对象类型，其与该属性相关的功能都应进行测试。

1. Out\_Of\_Service，Status\_Flags和Reliability测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款: 12.1.7， 12.1.9， 12.1.10， 12.2.7， 12.2.9， 12.2.10， 12.3.7， 12.3.9， 12.3.10， 12.4.6， 12.4.8，12.4.9，12.6.7， 12.6.9， 12.6.10， 12.7.7， 12.7.9， 12.7.10， 12.8.6， 12.8.8， 12.8.9， 12.15.8， 12.15.10， 12.15.11， 12.16.8， 12.16.10，12.16.11， 12.17.6， 12.17.8， 12.17.9， 12.18.7， 12.18.9， 12.18.10， 12.19.7， 12.19.9， 12.19.10， 12.20.6， 12.20.8， 12.20.9，12.23.7， 12.23.9， 和12.23.10。

目的：测试用例验证当Out\_Of\_Service为TRUE时Present\_Value是可写的。这也验证了Out\_Of\_Service，Status\_Flags和Reliability属性之间的关系。如果PICS表明在测试对象的 Out\_Of\_Service属性不可写并且如果该属性的值也不能通过其他方式来改变，则不进行此测试。这个测试适用于累加器，模拟输入，模拟输出，模拟量，模拟值，二进制输入，二进制输出，二进制值，生命安全点，生命安全区，多状态输入，多状态输出，多状态输出状态值，循环和脉冲变换器。

测试的概念：IUT在每个合适对象类型选择一个实例测试，若不支持可靠性属性则省略步骤4。

测试步骤：

1. IF (Out\_Of\_Service is writable) THEN

WRITE Out\_Of\_Service = TRUE

ELSE

MAKE (Out\_Of\_Service TRUE)

2. VERIFY Out\_Of\_Service = TRUE

3. VERIFY Status\_Flags=(?，FALSE，?，TRUE)

4. REPEAT X = (all values meeting the functional range requirements of 7.2.1) DO {

WRITE Present\_Value=X

VERIFY Present\_Value=X

}

5. IF (Reliability is writable) THEN

REPEAT X = (all values of the Reliability enumeration appropriate to the object type except NO\_FAULT\_DETECTED) DO {

WRITE Reliability = X

VERIFY Reliability = X

VERIFY Status\_Flags = (?，TRUE，?，TRUE)

WRITE Reliability = NO\_FAULT\_DETECTED

VERIFY Reliability = NO\_FAULT\_DETECTED

VERIFY Status\_Flags = (?，FALSE，?，TRUE)

}

6. IF (Out\_Of\_Service is writable) THEN

WRITE Out\_Of\_Service = FALSE

ELSE

MAKE (Out\_Of\_Service FALSE)

7. VERIFY Out\_Of\_Service = FALSE

8. VERIFY Status\_Flags = (?, ?, ?, FALSE)

注释：如果被测试对象是可命令的，且有一个内部程序可写入Present\_Value属性，则每个WriteProperty请求应包含一个足以覆盖内部进程的优先级。步骤4以后应舍弃优先级数组时隙。

1. 放弃默认测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.3.16，12.4.12，12.7.22，12.8.20，12.19.14，12.20.13。

目的：当舍弃了所有优先级的命令时，验证Present\_Value属性采用Relinquish\_Default的值。本测试适用于模拟量输出、模拟量、二进制值输出、二进制值、多输出、多状态值对象的控制。

测试思想：测试的前提条件是需提供一个丢弃了全部优先级命令，且考虑了任何最短开/关时间的对象。比较Present\_Value与Relinquish\_Default的值确保相同。如果可能的话，改变Relinquish\_Default值来验证Present\_Value的改变。

配置要求：配置被测试对象使其在Priority\_Array内所有时隙值为空，且没有内部算法向此对象发出优先级命令。

测试步骤：

1. VERIFY Priority\_Array=(NULL， NULL， NULL， NULL， NULL， NULL， NULL， NULL，NULL， NULL， NULL，NULL， NULL， NULL， NULL， NULL)

2. TRANSMIT ReadProperty-Request，

'Object Identifier' = (the object being tested)，

'Property Identifier' = Present\_Value

3. RECEIVE ReadProperty-ACK，

'Object Identifier' = (the object being tested)，

'Property Identifier' = Present\_Value

'Property Value' = (any valid value，X)

4. VERIFY Relinquish\_Default = X

5. IF (Relinquish\_Default is writable) THEN

WRITE Relinquish\_Default = (any valid value，Y，other than the one returned in step 3)

VERIFY Present\_Value = Y

1. 命令优先级测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：19.2。

目的：验证命令优先级算法的正确执行。该测试适用于模拟输出、模拟值、二进制输出、二进制值、多状态输出和可命令的多状态值对象。

测试思想：TD从5.4.2节指定的有效值中选择三个不同的值：Vlow，Vmed，和Vhigh。二进制数据类型Vlow 和 Vhigh应是相同的，Vmed不同。TD还选择了三个在1和5之间的优先级Plow， Pmed 和 Phigh，这些值Plow > Pmed > Phigh。选定的值按相应的优先级一次写入一个Present\_Value。检查Present\_Value和Priority\_Array来验证操作是否正确。小于6的优先级（优先级更高）用于考虑消除对最小开/关时间因素。

配置要求：配置被测对象，使优先级大于6的Priority\_Array所有时隙为空值。

测试步骤：

1. WRITE Present\_Value = Vlow ，PRIORITY = Plow

2. VERIFY Present\_Value = Vlow

3. VERIFY Priority\_Array = Vlow ，ARRAY INDEX = Plow

4. REPEAT Z = (each index 1 through 5 not equal to Plow ) DO {

VERIFY Priority\_Array = NULL，ARRAY INDEX = Z

}

5. WRITE Present\_Value = Vhigh ，PRIORITY = Phigh

6. VERIFY Present\_Value = Vhigh

7. VERIFY Priority\_Array = Vhigh ，ARRAY INDEX = Phigh

8. REPEAT Z = (each index 1 through 5 not equal to Plow or Phigh ) DO {

VERIFY Priority\_Array = NULL，ARRAY INDEX = Z

}

9. WRITE Present\_Value = Vmed ，PRIORITY = Pmed

10. VERIFY Present\_Value = Vhigh

11. VERIFY Priority\_Array = Vmed ，ARRAY INDEX = Pmed

12. REPEAT Z = (each index 1 through 5 not equal to Plow ，Pmed or Phigh ) DO {

VERIFY Priority\_Array = NULL，ARRAY INDEX = Z

}

13. WRITE Present\_Value = NULL，PRIORITY = Phigh

14. VERIFY Present\_Value = Vmed

15. REPEAT Z = (each index 1 through 5 not equal to Plow or Pmed ) DO {

VERIFY Priority\_Array = NULL，ARRAY INDEX = Z

}

16. WRITE Present\_Value = NULL，PRIORITY = Pmed

17. VERIFY Present\_Value = Vlow

18. REPEAT Z = (each index 1 through 5 not equal to Plow ) DO {

VERIFY Priority\_Array = NULL，ARRAY INDEX = Z

}

19. WRITE Present\_Value = NULL，PRIORITY = Plow

20. REPEAT Z = (each index 1 through 5) DO {

VERIFY Priority\_Array = NULL，ARRAY INDEX = Z

}

1. 最小关断时间

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.7.19，12.8.17，19.2.3，附件一。

目的：验证最小关断时间算法的实现。若不支持最小关断时间则跳过该测试。本测试适用于二进制输出和二进制值。

测试思想：设置测试对象的初始Present\_Value为ACTIVE，其优先级数值超过6（数值越高，优先级别越低）。对象在此状态的时间足够长使得所有最小关断时间都已过期。在Present\_Value写入优先级为7的INACTIVE值。监测Priority\_Array时隙6的值，验证其在最小关断时间内有值INACTIVE。

配置要求：配置被测对象，使优先级小于7的Priority\_Array所有时隙值为空，Present\_Value值是ACTIVE，并且内部算法不能向对象发出比当前优先级高的优先级命令。

测试步骤：

1. WRITE Present\_Value = INACTIVE，PRIORITY = 7

2. VERIFY Present\_Value = INACTIVE

3. VERIFY Priority\_Array = INACTIVE，ARRAY\_INDEX = 6

4. WAIT (approximately 90% of Minimum\_Off\_Time from step 1)

5. VERIFY Priority\_Array = INACTIVE，ARRAY\_INDEX = 6

6. WAIT **(Minimum ON/OFF Fail Time** + Minimum\_Off\_Time from step 1)

7. VERIFY Priority\_Array = NULL，ARRAY INDEX = 6

1. 最小开启时间

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.7.20，12.8.18，19.2.3，附件一。

目的：验证最小时间算法的执行。若不支持最小开启时间则跳过该测试。本测试适用于二进制输出和二进制值。

概念测试：设置测试对象的初始Present\_Value为INACTIVE，并且它被控制在数值上大于(优先级低于)6的优先级。对象在此状态的时间足够长使得所有最小开启时间都已过期。Present\_Value写入优先级为7的ACTIVE值。监测Priority\_Array时隙6的值，验证其在最小开启时间内是否有值ACTIVE。

配置要求：配置被测对象，使优先级数组中小于7的Priority\_Array所有时隙值为空，Present\_Value是INACTIVE，并且内部算法不能向对象发出比当前优先级高的优先级命令。

测试步骤：

1. WRITE Present\_Value = ACTIVE，PRIORITY = 7

2. VERIFY Present\_Value = ACTIVE

3. VERIFY Priority\_Array = ACTIVE，ARRAY\_INDEX = 6

4. WAIT (approximately 90% of Minimum\_On\_Time from step 1)

5. VERIFY Priority\_Array = ACTIVE，ARRAY\_INDEX = 6

6. WAIT (**Minimum ON/OFF Fail Time** + Minimum\_On\_Time from step 1)

7. VERIFY Priority\_Array = NULL， ARRAY INDEX = 6

1. 最小覆盖时间

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.7.19,12.7.20,12.8.17，12.8.18，19.2。

目的：验证更高优先级命令覆盖或关闭时间的最小值。如果最小开启时间和最小关断时间都不支持，则跳过此测试。本测试适用于二进制输出和二进制值的对象。

测试思想：测试对象的初始Present\_Value设置为INACTIVE，其优先级数值大于6。对象在此状态的时间足够长使得所有最小开启时间都已过期。在Present\_Value写入优先级为7的ACTIVE值。监测Priority\_Array时隙6的值，验证其包含值ACTIVE。在最小开启时间到期前，在Present\_Value写入优先级数值小于6的INACTIVE。这将使最小开启时间被覆盖，并且立即启动最小关断时间算法。

配置要求：配置被测对象，使优先级小于7的Priority\_Array所有时隙值为空，并且内部算法不能向对象发出比当前优先级数值小（优先级别高）的优先级命令。

测试步骤：

1. WRITE Present\_Value = ACTIVE，PRIORITY = 7

2. VERIFY Present\_Value = ACTIVE

3. VERIFY Priority\_Array = ACTIVE，ARRAY\_INDEX = 6

4. BEFORE Minimum\_On\_Time

WRITE Present\_Value = INACTIVE，PRIORITY = (any value numerically lower than 6 (higher priority))

5. VERIFY Present\_Value = INACTIVE

6. VERIFY Priority\_Array = INACTIVE，PRIORITY = 6

注释：如果支持最小关断时间，不支持最小开启时间，则测试步骤1至3的ACTIVE改为INACTIVE，步骤4至7的INACTIVE改为ACTIVE，步骤4仍用最小关断时间。

1. COV测试

8.2和9.6中包括了执行COV功能的测试。

1. 状态测试中二进制值对象的变化

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.6.14，12.6.15，12.6.16，12.7.14，12.7.15，12.7.16，12.8.12，12.8.13，和12.8.14。

目的：验证二进制对象所有属性是否随跟踪状态的更改（present\_value的改变）而更改。若不支持Change\_Of\_State\_Count， Change\_Of\_State\_Time和 Time\_Of\_State\_Count\_Reset属性，则跳过此测试。本测试适用于二进制输入、二进制输出和二进制值对象。

测试思想：改变测试二进制对象的Present\_Value。查看Change\_Of\_State\_Count属性值，检测其是否增加，查看Change\_Of\_State\_Time属性值，检测其是否被更新。复位Change\_Of\_State\_Count，查看Time\_Of\_State\_Count\_Reset属性值，检测其是否已更新。

配置要求：配置被测试的对象，使其Present\_Value和Change\_Of\_State\_Count属性可写或可更改。

测试步骤：

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = (the object being tested),

'Property Identifier' = Present\_Value

2. RECEIVE ReadProperty-ACK,

'Object Identifier' = (the object being tested),

'Property Identifier' = Present\_Value,

'Property Value' = ACTIVE | INACTIVE

3. TRANSMIT ReadProperty-Request,

'Object Identifier' = (the object being tested),

'Property Identifier' = Change\_Of\_State\_Count

4. RECEIVE ReadProperty-ACK,

'Object Identifier' = (the object being tested),

'Property Identifier' = Change\_Of\_State\_Count,

'Property Value' = (any valid value, N)

5. IF (Present\_Value is writable) THEN

IF (the value returned in step 2 was ACTIVE) THEN

WRITE Present\_Value = INACTIVE

VERIFY Present\_Value = INACTIVE

ELSE

WRITE Present\_Value = ACTIVE

VERIFY Present\_Value = ACTIVE

ELSE

MAKE (Present\_Value change to the opposite state)

6. TRANSMIT ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Local\_Date

7. RECEIVE ReadProperty-ACK,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Local Date,

'Property Value' = (the current local date, D)

8. TRANSMIT ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Local\_Time

9. RECEIVE ReadProperty-ACK,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Local\_Time,

'Property Value' = (the current local time, TLOC)

10. WAIT **Internal Processing Fail Time**

11. TRANSMIT ReadProperty-Request,

'Object Identifier' = (the object being tested),

'Property Identifier' = Change\_Of\_State\_Time

12. RECEIVE ReadProperty-ACK,

'Object Identifier' = (the object being tested),

'Property Identifier' = Change\_Of\_State\_Time,

'Property Value' = (a date and time such that the date = D and the time is approximately TLOC)

13. TRANSMIT ReadProperty-Request,

'Object Identifier' = (the object being tested),

'Property Identifier' = Change\_Of\_State\_Count

14. RECEIVE ReadProperty-ACK,

'Object Identifier' = (the object being tested),

'Property Identifier' = Change\_Of\_State\_Count,

'Property Value' = N + 1

15. IF (Change\_Of\_State\_Count is writable) THEN

WRITE Change\_Of\_State\_Count = 0

VERIFY Change\_Of\_State\_Count = 0

ELSE

MAKE (Change\_Of\_State\_Count = 0)

16. TRANSMIT ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Local\_Time

17. RECEIVE ReadProperty-ACK,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Local\_Time,

'Property Value' = (the current local time, TLOC)

18. TRANSMIT ReadProperty-Request,

'Object Identifier' = (the object being tested),

'Property Identifier' = Time\_Of\_State\_Count\_Reset

19. RECEIVE ReadProperty-ACK,

'Object Identifier' = (the object being tested),

'Property Identifier' = Time\_Of\_State\_Count\_Reset,

'Property Value' = (a date and time such that the date = D and the time is approximately TLOC)

1. 二进制对象经过的活动时间测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.6.17，12.6.18，12.7.17，12.7.18，12.8.15，12.8.16。

目的：验证二进制对象的全部属性是否跟踪活动时间而变化。若不支持Elapsed\_Active\_Time和Time\_Of\_Active\_Time\_Reset属性则跳过此试验。本测试适用于二进制输入、二进制输出和二进制值对象。

测试思想：设置二进制对象的Present\_Value为INACTIVE。查看Elapsed\_Active\_Time属性，验证当对象处于INACTIVE状态时不累积时间。然后设置Present\_Value为ACTIVE，查看Elapsed\_Active\_Time的属性，验证当对象处于ACTIVE时累积时间。再设置Present\_Value为INACTIVE，复位Elapsed\_Active\_Time。查看Time\_Of\_Active\_Time\_Reset，验证其已被更新。

配置要求：配置测试对象，使得Present\_Value和Elapsed\_Active\_Time属性可写或可改变。

测试步骤：

1. IF (Present\_Value is writable) THEN

WRITE Present\_Value = INACTIVE

VERIFY Present\_Value = INACTIVE

ELSE

MAKE (Present\_Value = INACTIVE)

2. TRANSMIT ReadProperty-Request，

'Object Identifier' = (the object being tested),

'Property Identifier' = Elapsed\_Active\_Time

3. RECEIVE ReadProperty -ACK，

'Object Identifier' = (the object being tested),

'Property Identifier' = Elapsed\_Active\_Time,

'Property Value' = (the elapsed active time, TELAPSED in seconds)

4. WAIT (1 minute)

5. TRANSMIT ReadProperty -Request，

'Object Identifier' = (the object being tested),

'Property Identifier' = Elapsed\_Active\_Time

6. RECEIVE ReadProperty -ACK，

'Object Identifier' = (the object being tested),

'Property Identifier' = Elapsed\_Active\_Time,

'Property Value' = (the same TELAPSED as step 3)

7. IF (Present\_Value is writable) THEN

WRITE Present\_Value = ACTIVE

VERIFY Present\_Value = ACTIVE

ELSE

MAKE (Present\_Value = ACTIVE)

8. WAIT (**Internal Processing Fail Time** + 30 seconds)

9. TRANSMIT ReadProperty -Request，

'Object Identifier' = (the object being tested),

'Property Identifier' = Elapsed\_Active\_Time

10. RECEIVE ReadProperty -ACK，

'Object Identifier' = (the object being tested),

'Property Identifier' = Elapsed\_Active\_Time,

'Property Value' = (T: (TELAPSED + 30) ≤ T ≤ (TELAPSED + 30 + **Internal Processing Fail Time**))

11. IF (Present\_Value is writable) THEN

WRITE Present\_Value=INACTIVE

VERIFY Present\_Value=INACTIVE

ELSE

MAKE (Present\_Value=INACTIVE)

12. IF (Elapsed\_Active\_Time is writable) THEN

WRITE Elapsed\_Active\_Time=0

VERIFY Elapsed\_Active\_Time=0

ELSE

MAKE (Elapsed\_Active\_Time=0)

13. TRANSMIT ReadProperty -Request，

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Local\_Date

14. RECEIVE ReadProperty -ACK，

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Local Date,

'Property Value' = (the current local date, D)

15. TRANSMIT ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Local\_Time

16. RECEIVE ReadProperty-ACK,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Local\_Time,

'Property Value' = (the current local time, TLOC)

17. TRANSMIT ReadProperty-Request,

'Object Identifier' = (the object being tested),

'Property Identifier' = Time\_Of\_Active\_Time\_Reset

18. RECEIVE ReadProperty-ACK,

'Object Identifier' = (the object being tested),

'Property Identifier' = Present\_Value,

'Property Value' = (a date and time such that the date = D and the time is approximately TLOC)

1. Event\_Enable测试

所需测试：ConfirmedEventNotification服务启动测试，9.4；UnconfirmedEventNotification服务启动测试，9.5；ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.1.27，12.2.23，12.3.24，12.4.20，12.6.22，12.7.26，12.8.24，12.12.10，12.15.19，12.16.19，12.17.33，12.18.17，12.110.18，12.20.18，12.23.26，和12.25.22。

目的：证明只有Event\_Enable与事件转变的对应位值为TURE时，通知消息才能被发送。本测试适用于事件注册对象和累加器、模拟输入、模拟输出、模拟值、二进制输入、二进制输出、二进制值、生命安全点、生命安全区、循环、多态输入、多态输出、多状态值、脉冲转换器和支持内部报告的趋势日志对象。

测试思想：设置IUT，使Event\_Enable属性说明哪些事件变化会触发事件通知，哪些不会。然后触发所有事件转换，查看IUT，验证只有事件使能属性值为TRUE时才会发送通知消息。

配置要求：设置Event\_Enable属性，TO-OFFNORMAL发生变化和TO-NORMAL发生改变的属性值都为TRUE，其他事件转换属性值为FALSE。对于模拟对象，设置Limit\_Enable属性为（TRUE，TRUE）。引用事件触发属性应设为可以引起NORMAL状态的值。所有接收端的转换参数值都设为（TRUE，TRUE，TRUE）。

以下为测试描述，其中“X”指触发事件的属性。

1. VERIFY Event\_State = NORMAL

2. WAIT (Time\_Delay + **Notification Fail Time**)

3. IF (X is writable) THEN

WRITE X = (a value that is OFFNORMAL)

ELSE

MAKE (X have a value that is OFFNORMAL)

4. WAIT (Time\_Delay)

5. BEFORE **Notification Fail Time**

IF (the Transitions bit corresponding to the TO-OFFNORMAL transition is TRUE)THEN

RECEIVE ConfirmedEventNotification-Request,

'Process Identifier' = (any valid process ID),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the event-generating object configured for this test),

'Time Stamp' = (the current local time),

'Notification Class' = (the class corresponding to the object being tested),

'Priority' = (the value configured to correspond to a TO-

OFFNORMAL transition),

'Event Type' = (any valid event type),

'Notify Type' = EVENT | ALARM,

'AckRequired' = TRUE | FALSE,

'From State' = NORMAL,

'To State' = OFFNORMAL,

'Event Values' = (values appropriate to the event type)

ELSE

CHECK (verify that the IUT did not transmit an event notification message)

6. VERIFY Event\_State = OFFNORMAL

7. IF (X is writable) THEN

WRITE X = (a value that is NORMAL)

ELSE

MAKE (X have a value that is NORMAL)

8. WAIT (Time\_Delay)

9. BEFORE **Notification Fail Time**

IF (the Transitions bit corresponding to the TO-NORMAL transition is TRUE) THEN

RECEIVE ConfirmedEventNotification-Request,

'Process Identifier' = (any valid process ID),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the event-generating object configured for this test),

'Time Stamp' = (the current local time),

'Notification Class' = (the class corresponding to the object being tested),

'Priority' = (the value configured to correspond to a TO-NORMAL

transition),

'Event Type' = (any valid event type),

'Notify Type' = EVENT | ALARM,

'AckRequired' = TRUE | FALSE,

'From State' = OFFNORMAL,

'To State' = NORMAL,

'Event Values' = (values appropriate to the event type)

ELSE

CHECK (verify that the IUT did not transmit an event notification message)

10. VERIFY Event\_State = NORMAL

11. IF (the event-triggering object can be placed into a fault condition) THEN {

MAKE (the event-triggering object change to a fault condition)

BEFORE **Notification Fail Time**

IF (the Transitions bit corresponding to the TO-FAULT transition is TRUE) THEN

RECEIVE ConfirmedEventNotification-Request,

'Process Identifier' = (any valid process ID),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the event-generating object configured for this test),

'Time Stamp' = (the current local time),

'Notification Class' = (the class corresponding to the object being tested),

'Priority' = (the value configured to correspond to a TO-

FAULT transition),

'Event Type' = (any valid event type),

'Notify Type' = EVENT | ALARM,

'AckRequired' = TRUE | FALSE,

'From State' = NORMAL,

'To State' = FAULT,

'Event Values' = (values appropriate to the event type)

ELSE

CHECK (verify that the IUT did not transmit an event notification message)

VERIFY Event\_State = FAULT

}

注释：UnconfirmedEventNotification服务可以取代ConfirmedEventNotification服务。在测试描述中，由于‘Message Text’参数可选因此被省略。IUT通知消息中可能包括此参数。

1. Acked\_Transitions测试

所需测试：ConfirmedEventNotification服务启动测试，9.4；UnconfirmedEventNotification服务启动测试，9.5；AcknowledgeAlarm服务执行测试，10.1；ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.1.28，12.2.24，12.3.25，12.4.21，12.6.23，12.7.27，12.8.25，12.12.11，12.15.20，12.16.20， 12.17.35，12.18.18，12.19.19，12.20.19，12.23.27，12.25.23。

目的：验证Acked\_Transitions属性是否跟踪已收到一个先前发出的事件通知。同时验证Status\_Flags 和Event\_State之间的关系。此测试适用于事件注册对象和累加器、模拟输入、模拟输出、模拟值、二进制输入、二进制输出、二进制值、生命安全点、生命安全区、循环、多状态输入、多状态输出、多状态值、脉冲转换器和支持内部报告的趋势日志对象。

测试思想：配置IUT的Event\_Enable属性，说明所有事件变化都可以触发事件通知。Acked\_Transitions属性值为（TRUE，TRUE，TRUE），表明所有先前的转换已被确认。触发所有事件变化，监测Acked\_Transitions属性，验证如果收到确认一个通知消息被发送和重置，则属性值相应的位被清除。

配置要求：The Event\_Enable 和Acked\_Transitions属性应设置为值（TRUE，TRUE，TRUE）。对于模拟对象，Limit\_Enable属性应设置为值（TRUE，TRUE）。引用的事件触发属性应设置为导致NORMAL状态的值，所有接收方的Transitions参数值应为（TRUE，TRUE，TRUE）

在下面的测试描述中，“X”指事件触发属性。

测试步骤：

1. WAIT (Time\_Delay + **Notification Fail Time**)

2. VERIFY Event\_State = NORMAL

3. VERIFY Acked\_Transitions = (TRUE，TRUE，TRUE)

4. VERIFY Status\_Flags = (FALSE，FALSE，?，?)

5. IF (X is writable) THEN

WRITE X = (a value that is OFFNORMAL)

ELSE

MAKE (X have a value that is OFFNORMAL)

6. WAIT (Time\_Delay)

7. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request，

'Process Identifier' = (any valid process ID),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the event-generating object configured for this test),

'Time Stamp' = (any valid time stamp),

'Notification Class' = (the class corresponding to the object being tested),

'Priority' = (the value configured to correspond to a TO-OFFNORMAL

transition),

'Event Type' = (any valid event type),

'Notify Type' = (the notify type configured for this event),

'AckRequired' = TRUE,

'From State' = NORMAL,

'To State' = OFFNORMAL,

'Event Values' = (values appropriate to the event type)

8. TRANSMIT BACnet-SimpleACK-PDU

9. VERIFY Event\_State = OFFNORMAL

10. VERIFY Acked\_Transitions = (FALSE，TRUE，TRUE)

11. VERIFY Status\_Flag = (TRUE，FALSE，?，?)

12. IF (X is writable) THEN

WRITE X = (a value that is NORMAL)

ELSE

MAKE (X have a value that is NORMAL)

13. WAIT (Time\_Delay)

14. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request，

'Process Identifier' = (any valid process ID),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the event-generating object configured for this test),

'Time Stamp' = (any valid time stamp),

'Notification Class' = (the class corresponding to the object being tested),

'Priority' = (the value configured to correspond to a TO-NORMAL transition),

'Event Type' = (any valid event type),

'Notify Type' = (the notify type configured for this event),

'AckRequired' = TRUE,

'From State' = OFNORMAL,

'To State' = NORMAL,

'Event Values' = (values appropriate to the event type)

15. TRANSMIT BACnet-SimpleACK-PDU

16. VERIFY Event\_State = NORMAL

17. VERIFY Acked\_Transitions = (FALSE，TRUE，FALSE)

18. VERIFY Status\_Flags = (FALSE，FALSE，?，?)

19. IF (the event-triggering object can be placed into a fault condition) THEN

20. MAKE (the event-triggering object change to a fault condition)

21. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request，

'Process Identifier' = (any valid process ID),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the event-generating object configured for this test),

'Time Stamp' = (any valid time stamp),

'Notification Class' = (the class corresponding to the object being tested),

'Priority' = (the value configured to correspond to a TO-FAULT

transition),

'Event Type' = (any valid event type),

'Notify Type' = (the notify type configured for this event),

'AckRequired' = TRUE,

'From State' = NORMAL,

'To State' = FAULT,

'Event Values' = (values appropriate to the event type)

22. TRANSMIT BACnet-SimpleACK-PDU

23. VERIFY Event\_State = FAULT

24. VERIFY Acked\_Transitions = (FALSE，FALSE，FALSE)

25. VERIFY Status\_Flags = (FALSE，TRUE，?，?)

26. MAKE (the event-triggering object change to a normal condition)

27. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request，

'Process Identifier' = (any valid process ID),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the event-generating object configured for this test),

'Time Stamp' = (any valid time stamp),

'Notification Class' = (the class corresponding to the object being tested),

'Priority' = (the value configured to correspond to a TO-NORMAL

transition),

'Event Type' = (any valid event type),

'Notify Type' = (the notify type configured for this event),

'AckRequired' = TRUE,

'From State' = FAULT,

'To State' = NORMAL,

'Event Values' = (values appropriate to the event type)

28. TRANSMIT BACnet-SimpleACK-PDU

29. VERIFY Event\_State = NORMAL

30. VERIFY Acked\_Transitions = (FALSE，FALSE，FALSE)

31. VERIFY Status\_Flags = (FALSE，FALSE，?，?)

32. TRANSMIT AcknowledgeAlarm-Request，

'Acknowledging Process Identifier' = (the value of the 'Process Identifier' in step 21),

'Event Object Identifier' = (the 'Event Object Identifier' in step 21),

'Event State Acknowledged' = FAULT,

'Time Stamp' = (the 'Time Stamp in step 21),

'Time of Acknowledgment' = (the TD’s current time)

33. RECEIVE BACnet-SimpleACK-PDU

34. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request，

'Process Identifier' = (the value of the 'Process Identifier' in step 21),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the 'Event Object Identifier' in step 21),

'Time Stamp' = (the 'Time Stamp' in step 21),

'Notification Class' = (the 'Notification Class' in step 21),

'Priority' = (the 'Priority' in step 21),

'Event Type' = (the 'Event Type' in step 21),

'Notify Type' = ACK\_NOTIFICATION,

'To State' = FAULT

ELSE

BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request，

'Process Identifier' = (the value of the 'Process Identifier' in step 21),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the 'Event Object Identifier' in step 21),

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the 'Notification Class' in step 21),

'Priority' = (the 'Priority' in step 21),

'Event Type' = (the 'Event Type' in step 21),

'Notify Type' = ACK\_NOTIFICATION

35. TRANSMIT BACnet-SimpleACK-PDU

36. VERIFY Acked\_Transitions = (FALSE，TRUE，FALSE)

37. TRANSMIT AcknowledgeAlarm-Request，

'Acknowledging Process Identifier' = (the value of the 'Process Identifier' in step 27),

'Event Object Identifier' = (the 'Event Object Identifier' in step 27),

'Event State Acknowledged' = NORMAL,

'Time Stamp' = (the 'Time Stamp in step 27),

'Time of Acknowledgment' = (the TD’s current time)

38. RECEIVE BACnet-SimpleACK-PDU

39. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request，

'Process Identifier' = (the value of the 'Process Identifier' in step 27),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the 'Event Object Identifier' in step 27),

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the 'Notification Class' in step 27),

'Priority' = (the 'Priority' in step 27),

'Event Type' = (the 'Event Type' in step 27),

'Notify Type' = ACK\_NOTIFICATION,

'To State' = NORMAL

ELSE

BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request，

'Process Identifier' = (the value of the 'Process Identifier' in step 27),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the 'Event Object Identifier' in step 27),

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the 'Notification Class' in step 27),

'Priority' = (the 'Priority' in step 27),

'Event Type' = (the 'Event Type' in step 27),

'Notify Type' = ACK\_NOTIFICATION

40. TRANSMIT BACnet-SimpleACK-PDU

41. VERIFY Acked\_Transitions = (FALSE，TRUE，TRUE)

42. TRANSMIT AcknowledgeAlarm-Request，

'Acknowledging Process Identifier' = (the value of the 'Process Identifier' in step 7),

'Event Object Identifier' = (the 'Event Object Identifier' in step 7),

'Event State Acknowledged' = OFFNORMAL,

'Time Stamp' = (the 'Time Stamp in step 7),

'Time of Acknowledgment' = (the TD’s current time)

43. RECEIVE BACnet-SimpleACK-PDU

44. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request，

'Process Identifier' = (the value of the 'Process Identifier' in step 7),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the 'Event Object Identifier' in step 7),

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the 'Notification Class' in step 7),

'Priority' = (the 'Priority' in step 7),

'Event Type' = (the 'Event Type' in step 7),

'Notify Type' = ACK\_NOTIFICATION,

'To State' = OFFNORMAL

ELSE

BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request，

'Process Identifier' = (the value of the 'Process Identifier' in step 7),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the current time or sequence number),

'Time Stamp' = (the 'Time Stamp' in step 7),

'Notification Class' = (the 'Notification Class' in step 7),

'Priority' = (the 'Priority' in step 7),

'Event Type' = (the 'Event Type' in step 7),

'Notify Type' = ACK\_NOTIFICATION

45.TRANSMIT BACnet-SimpleACK-PDU

46.VERIFY Acked\_Transitions = (TRUE，TRUE，TRUE)

注释：UnconfirmedEventNotification服务可代替ConfirmedEventNotification服务。在测试说明中，‘Message Text’参数因可选，故被省略。IUT通知消息中可能包括此参数。

1. Notify\_Type测试

所需测试：ConfirmedEventNotification服务启动测试，9.4；UnconfirmedEventNotification服务启动测试，9.5；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.1.29，12.2.25，12.3.26，12.4.22，12.6.24，12.7.28，12.8.26，12.12.6，12.15.21，12.16.21，12.17.36，12.18.19，12.19.20，12.20.20，12.23.28和12.25.24。

目的：验证Notify\_Type属性决定事件通知是作为警告被发送还是作为事件被发送。本测试适用于事件注册对象、累加器、模拟输入、模拟输出、模拟值、二进制输入、二进制输出、二进制值、生命安全点、生活安全区、循环、多状态输入、多状态输出、多状态值、脉冲转换器和支持内部报告的趋势日志对象。

配置要求：用两个事件生成对象E1和E2配置IUT。对象E1应配置警报Notify\_Type，E2应配置事件Notify\_Type。在开始测试时，两个对象的事件状态都应为NORMAL，Event\_Enable和Acked\_Transitions属性值都为（TRUE，TRUE，TRUE）。模拟对象的Limit\_Enable属性值设为（TRUE，TRUE）。所有接收端的转换参数值都设为（TRUE，TRUE，TRUE）。

以下内容为测试步骤，其中X1和X2分别指事件E1和E2的触发属性。

测试步骤：

1. VERIFY (E1)，Event\_State=NORMAL

2. VERIFY (E2)，Event\_State=NORMAL

3. WAIT (Time\_Delay + **Notification Fail Time**)

4. IF (X1 is writable) THEN

WRITE X1=(a value that is OFFNORMAL)

ELSE

MAKE (X1 a value that is OFFNORMAL)

5. WAIT (Time\_Delay)

6. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request，

'Process Identifier' = (any valid process ID),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (E1),

'Time Stamp' = (the current local time),

'Notification Class' = (the class corresponding to the object being tested),

'Priority' = (the value configured to correspond to a TO- OFFNORMAL transition),

'Event Type' = (any valid event type),

'Notify Type' = ALARM,

'AckRequired' = TRUE | FALSE,

'From State' = NORMAL,

'To State' = OFFNORMAL,

'Event Values' = (values appropriate to the event type)

7. IF (X2 is writable) THEN

WRITE X2=(a value that is OFFNORMAL)

ELSE

MAKE (X2 a value that is OFFNORMAL)

8. WAIT (Time\_Delay)

9. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request，

'Process Identifier' = (any valid process ID),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (E2),

'Time Stamp' = (the current local time),

'Notification Class' = (the class corresponding to the object being tested),

'Priority' = (the value configured to correspond to a TO- OFFNORMAL transition),

'Event Type' = (any valid event type),

'Notify Type' = EVENT,

'AckRequired' = TRUE | FALSE,

'From State' = NORMAL,

'To State' = OFFNORMAL,

'Event Values' = (values appropriate to the event type)

注释：如果Notify\_Type可写，则可以通过将Notify\_Type从ALARM更改为EVENT，只使用一个事件生成对象来执行该测试，从而同时涵盖这两种情况。由于‘Message Text’参数可选，因此在测试描述将其忽略。IUT通知消息中可能包含此参数。

1. 限制使能测试

所需测试：ConfirmedEventNotification服务启动测试，9.4；UnconfirmedEventNotification服务启动测试，9.5；ReadProperty服务测试执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款: 12.1.26, 12.2.22, 12.3.23, 12.4.19 ，12.23.25。

目的：验证Limit\_Enable限制使能属性是否正确启用或禁用超出范围事件的报告。本测试适用于累加器，模拟输入，模拟输出，模拟值和本报告支持的脉冲转换器对象。如果Limit\_Enable属性不可写且无法重新配置，则跳过此测试。

测试思想：修改事件触发属性，使Limit\_Enable的每个可能值组合都超过上下限。查看生成的事件通知消息，验证只有在开启相关事件限制的情况下才会传输。

配置要求：设置Limit\_Enable属性值为（TRUE，TRUE）。如果Limit\_Enable不可写，应该有其他具有相同类型的限制使能值为（FALSE，TRUE），（TRUE，FALSE）和（FALSE，FALSE）的事件生成对象。

下述测试步骤中，“X”指事件触发属性。

1. VERIFY Limit\_Enable = (TRUE, TRUE)

2. VERIFY Event\_State = NORMAL

3. WAIT (Time\_Delay + **Notification Fail Time**)

4. IF (X is writable) THEN

WRITE X = (a value that exceeds High\_Limit)

ELSE

MAKE (X a value that exceeds High\_Limit)

5. WAIT (Time\_Delay)

6. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

'Process Identifier' = (any valid process ID),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object configured for this test),

'Time Stamp' = (the current local time),

'Notification Class' = (the class corresponding to the object being tested),

'Priority' = (the value configured to correspond to a TO- OFFNORMAL transition),

'Event Type' = OUT\_OF\_RANGE,

'Notify Type' = ALARM | EVENT,

'AckRequired' = TRUE | FALSE,

'From State' = NORMAL,

'To State' = HIGH\_LIMIT,

'Event Values' = (values appropriate to the event type)

7. IF (X is writable) THEN

WRITE X = (a value that is lower than Low\_Limit)

ELSE

MAKE (X a value that is lower than Low\_Limit)

8. WAIT (Time\_Delay)

9. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

'Process Identifier' = (any valid process ID),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object configured for this test),

'Time Stamp' = (the current local time),

'Notification Class' = (the class corresponding to the object being tested),

'Priority' = (the value configured to correspond to a TO- OFFNORMAL transition),

'Event Type' = OUT\_OF\_RANGE,

'Notify Type' = ALARM | EVENT,

'AckRequired' = TRUE | FALSE,

'From State' = HIGH\_LIMIT,

'To State' = LOW\_LIMIT,

'Event Values' = (values appropriate to the event type)

10. WRITE Limit\_Enable = (FALSE, TRUE)

11. VERIFY Limit\_Enable = (FALSE, TRUE)

12. IF (X is writable) THEN

WRITE X = (a value that exceeds High\_Limit)

ELSE

MAKE (X a value that exceeds High\_Limit)

13. WAIT (Time\_Delay)

14. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

'Process Identifier' = (any valid process ID),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object configured for this test),

'Time Stamp' = (the current local time),

'Notification Class' = (the class corresponding to the object being tested),

'Priority' = (the value configured to correspond to a TO- OFFNORMAL transition),

'Event Type' = OUT\_OF\_RANGE,

'Notify Type' = ALARM | EVENT,

'AckRequired' = TRUE | FALSE,

'From State' = LOW\_LIMIT,

'To State' = HIGH\_LIMIT,

'Event Values' = (values appropriate to the event type)

15. IF (X is writable) THEN

WRITE X = (a value that is lower than Low\_Limit)

ELSE

MAKE (X a value that is lower than Low\_Limit)

16. WAIT (Time\_Delay + **Notification Fail Time**)

17. CHECK (verify that no notification message was transmitted)

18. WRITE Limit\_Enable = (TRUE, FALSE)

19. VERIFY Limit\_Enable = (TRUE, FALSE)

20. IF (X is writable) THEN

WRITE X = (a value that exceeds High\_Limit)

ELSE

MAKE (X a value that exceeds High\_Limit)

21. WAIT (Time\_Delay + **Notification Fail Time**)

22. CHECK (verify that no notification message was transmitted)

23. IF (X is writable) THEN

WRITE X = (a value that is lower than Low\_Limit)

ELSE

MAKE (X a value that is lower than Low\_Limit)

24. WAIT (Time\_Delay)

25. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

'Process Identifier' = (any valid process ID),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object configured for this test),

'Time Stamp' = (the current local time),

'Notification Class' = (the class corresponding to the object being tested),

'Priority' = (the value configured to correspond to a TO- OFFNORMAL transition),

'Event Type' = OUT\_OF\_RANGE,

'Notify Type' = ALARM | EVENT,

'AckRequired' = TRUE | FALSE,

'From State' = HIGH\_LIMIT,

'To State' = LOW\_LIMIT,

'Event Values' = (values appropriate to the event type)

26. WRITE Limit\_Enable = (FALSE, FALSE)

27. VERIFY Limit\_Enable = (FALSE, FALSE)

28. IF (X is writable) THEN

WRITE X = (a value that exceeds High\_Limit)

ELSE

MAKE (X a value that exceeds High\_Limit)

29. WAIT (Time\_Delay + **Notification Fail Time**)

30. CHECK (verify that no notification message was transmitted)

31. IF (X is writable) THEN

WRITE X = (a value that is lower than Low\_Limit)

ELSE

MAKE (X a value that is lower than Low\_Limit)

32. WAIT (Time\_Delay + **Notification Fail Time**)

33. CHECK (verify that no notification message was transmitted)

注释：UnconfirmedEventNotification服务可代替ConfirmedEventNotification服务。测试描述中可省略“Message Text”参数，因为它是可选的。IUT通知消息中可包含此参数。如果Limit\_Enable属性不可写，则等效测试可应用于具有Limit\_Enable合适值的新对象或重新配置的对象。

1. 进程标识符测试
2. 进程标识属性测试

所需测试：无。

BACnet-2001参考条款：12.11.4。

目的：验证Process\_Identifier进程标识属性能否支持需要的值的范围。本测试只有当设备对象有Protocol\_Revision协议修订属性且其值为1到3时才执行。

测试思想：验证Process\_Identifier进程标识属性是否可以设置为从0到232-1范围内的任何值，方法是将其设置为最小值，最大值和中间的随机值。

1. IF the Process\_Identifier property is not writable THEN

MAKE (Process\_Identifier = 232-1)

ELSE

WRITE Process\_Identifier = 232-1

2. VERIFY Process\_Identifier = 232-1

3. IF the Process\_Identifier property is not writable THEN

MAKE (Process\_Identifier = 0)

ELSE

WRITE Process\_Identifier = 0

4. VERIFY Process\_Identifier = 0

5. IF the Process\_Identifier property is not writable THEN

MAKE (Process\_Identifier = (any value in the range 0.. 232-1))

ELSE

WRITE Process\_Identifier = (any value in the range 0.. 232-1)

6. VERIFY Process\_Identifier = (the value used in step 5)

1. 接收列表测试

所需测试：无。

BACnet参考条款：12.21.8。

目的：验证Recipient\_List属性中包含的Process\_Identifier是否支持所需的值的范围。

测试思想：通过将Recipient\_List属性中包含的Process\_Identifier设置为最小值0，最大值232-1和随机选择的中间值，验证其是否可以设置为0到232-1范围内的任何值。仅当设备对象中存在Protocol\_Revision属性且值大于等于1时，才进行此测试。

1. MAKE (an entry in the Recipient\_List = (any valid Valid\_Days, any valid time range, any valid)

Recipient, 232-1, TRUE | FALSE depending on the IUT support, any valid transitions bitstring)

2. VERIFY Process\_Identifier = 232-1

3. MAKE (an entry in the Recipient\_List = (any valid Valid\_Days, any valid time range, any valid Recipient, 0,TRUE | FALSE depending on the IUT support, any valid transitions bitstring))

4. VERIFY Process\_Identifier = 0

5. MAKE (an entry in the Recipient\_List = (any valid Valid\_Days, any valid time range, any valid

Recipient, any value in the range 0.. 232-1,TRUE | FALSE depending on the IUT support, any valid transitions Bitstring))

6. VERIFY Process\_Identifier = (the value used in step 5)

1. 状态数范围测试

所需测试：ReadProperty服务执行测试，8.1；WriteProperty服务执行测试，8.2.2。

BACnet参考条款：12.18.11,12.19.4,12.19.11,12.20.4,12.20.10。

目的：验证多状态输入，多状态输出或多状态值对象的当前值是否可以通过其Number\_Of\_States属性被限制在1内。

测试思想：首先验证Number\_Of\_States属性非零，修改引用对象的Present\_Value属性使其值大于Number\_Of\_States属性值，将Present\_Value属性设置为0，验证每种情况是否正确的做出错误响应。

配置要求：配置IUT，使其包含多状态输入、多状态输出、多状态值对象Object1，该对象中包含可写的Present\_Value属性。配置Object1，使其在测试执行之前Present\_Value属性可写。如果不能使Present\_Value属性可写，则跳过此测试。

测试步骤：

1. READ COUNT = Number\_Of\_States

2. VERIFY Number\_Of\_States = (a non-zero value)

3. TRANSMIT WriteProperty-Request,

'Object Identifier' = Object1,

'Property Identifier' = Present\_Value,

'Property Value' = (any value larger than COUNT)

4. RECEIVE BACnet-Error-PDU,

Error Class = PROPERTY,

Error Code = VALUE\_OUT\_OF\_RANGE

5. VERIFY (Object1), P1 = (a value between 1 and COUNT, inclusive)

6. TRANSMIT WriteProperty-Request,

'Object Identifier' = Object1,

'Property Identifier' = Present\_Value,

'Property Value' = 0

7. RECEIVE BACnet-Error-PDU,

Error Class = PROPERTY,

Error Code = VALUE\_OUT\_OF\_RANGE

8. VERIFY (Object1), P1 = (a value between 1 and COUNT, inclusive)

注释：在测试可命令对象时，可以任意选择优先级。

1. 指定对象测试

本节测试仅适用于指定的对象类型，并仅对每个支持对象类型的一个实例进行测试。

1. 模拟输入对象测试
2. 输入跟踪测试

所需测试：ReadProperty服务执行测试，10.18。

BACnet参考条款：12.2.4。

目的：验证跟踪和表示模拟输入值的性能。

配置要求：IUT应连接至模拟输入，该模拟输入可在测试过程中进行外部控制。还应提供验证该值合理所需的任何范围信息。配置与此物理输入关联的模拟输入对象为Out\_Of\_Service = FALSE。

测试步骤：

1. MAKE (the real analog input take on a known value near the middle of the supported range)

2. VERIFY Present\_Value = (a value that corresponds to the known input signal)

3. MAKE (the real analog input take on a higher known value within the supported range)

4. VERIFY Present\_Value = (a value that corresponds to the known input signal)

5. MAKE (the real analog input take on a value lower that the one used in step 1)

6. VERIFY Present\_Value = (a value that corresponds to the known input signal)

1. Out\_Of\_Service，Status\_Flags和Reliability测试

验证Out\_Of\_Service属性的功能以及Status\_Flags，Reliability和Out\_Of\_Service之间关系的测试在8.3.1.1中已有过介绍。

1. 值变更测试

验证模拟输入对象的值报告功能变化的测试将在9.2.1,9.2.2,9.3.1和9.3.2中介绍。

1. 内在报告测试

验证模拟输入对象内在报告功能的测试将在9.4.6中介绍。

1. 模拟输出对象测试
2. 输出跟踪测试

所需测试：ReadProperty服务执行测试，10.22。

BACnet参考条款：12.3.4。

目的：验证表示和执行物理模拟输出的能力。

配置：IUT连接可用万用表观察的模拟输出，提供验证模拟输出值合理性所需的任何范围信息。配置与此物理输入关联的模拟输出对象为Out\_Of\_Service = FALSE。

测试步骤：

1. WRITE Present\_Value = (a value near the middle of the supported range),

PRIORITY = (a priority higher than any internal algorithms writing to this property)

2. VERIFY Present\_Value = (the value written in step 1)

3. CHECK (with a multimeter to verify that the physical output value corresponds to the value of the Present\_Value property)

4. WRITE Present\_Value = (a value near the high limit of the supported range),

PRIORITY = (a priority higher than any internal algorithms writing to this property)

5. VERIFY Present\_Value = (the value written in step 4)

6. CHECK (with a multimeter to verify that the physical output value corresponds to the value of the Present\_Value property)

7. WRITE Present\_Value = (a value near the low limit of the supported range),

PRIORITY = (a priority higher than any internal algorithms writing to this property)

8. VERIFY Present\_Value = (the value written in step 7)

9. CHECK (with a multimeter to verify that the physical output value corresponds to the value of the Present\_Value property)

1. Out\_Of\_Service，Status\_Flags和Reliability测试

验证Out\_Of\_Service属性的功能以及Status\_Flags，Reliability和Out\_Of\_Service之间关系的测试已在8.3.1.1中介绍。

1. 优先命令测试

验证模拟输出对象的优先级命令的测试已在8.3.1.2和8.3.1.3中介绍。

1. 值变更测试

验证模拟输出对象的值报告功能变化的测试将在9.2.1,9.2.2,9.3.1和9.3.2中介绍。

1. 内在报告测试

验证模拟输出对象内在报告功能的测试将在9.4.6中介绍。

1. 模拟值对象测试
2. Out\_Of\_Service，Status\_Flags和Reliability测试

验证Out\_Of\_Service属性功能以及Status\_Flags，Reliability和Out\_Of\_Service之间关系的测试已在8.3.1.1中介绍。

1. 优先命令测试

验证模拟值对象优先级命令的测试已在8.3.1.2和8.3.1.3中介绍。

1. 值变更测试

验证模拟值对象的值报告功能变化的测试将在9.2.1,9.2.2,9.3.1和9.3.2中介绍。

1. 内在报告测试

验证模拟值对象的内在报告功能的测试将在9.4.6中介绍。

11. Averaging对象测试

Averaging对象提供了一种监视采样属性所获得的平均值，最小值和最大值的方法。采样属性的数据类型可以是BOOLEAN，INTEGER，Unsigned，Enumerated或Real。每种数据类型的测试在本节中只进行一次。

所需测试：ReadProperty服务执行测试，10.18;WriteProperty服务执行测试，10.22。

BACnet参考条款：12.5。

1. 样本再初始化

目的：验证Averaging对象当Average\_Value、Object\_Property\_Reference、Maximum\_Value Window\_Samples或Window\_Interval发生变化时，是否正确重置Attempted\_Samples，Valid\_Samples，Minimum\_Value，Attempted\_Samples属性。

测试思想：设置IUT，监视Averaging对象的某些属性值。通过依次写入Attempted\_Samples，Object\_Property\_Reference，Window\_Interval和Window\_Samples来重新初始化采样。每次重新初始化后，TD都会暂停并验证新的采样是否已经开始。

配置要求：IUT应配置一个某些属性值可监视的Averaging对象。采样间隔应足够长，以使TD能够验证样品是否已正确重新初始化。

1. VERIFY Minimum\_Value = (a value x: -INF < x < INF),

2. VERIFY Average\_Value = (a value ≠ NaN),

3. VERIFY Maximum\_Value = (a value x: Minimum\_Value ≤ x < INF),

4. VERIFY Attempted\_Samples = (a value x > 0),

5. VERIFY Valid\_Samples = (a value x > 0),

6. WRITE Attempted\_Samples = 0,

7. VERIFY Attempted\_Samples = 0,

8. VERIFY Minimum\_Value = INF,

9. VERIFY Maximum\_Value = -INF,

10. VERIFY Average\_Value = NaN,

11. VERIFY Valid\_Samples = 0,

12. WAIT (at least two sample times),

13. VERIFY Minimum\_Value = (a value x: -INF < x < INF),

14. VERIFY Average\_Value = (a value ≠ NaN),

15. VERIFY Maximum\_Value = (a value x: Minimum\_Value ≤ x < INF),

16. VERIFY Attempted\_Samples = (a value x ≥ 2),

17. VERIFY Valid\_Samples = (a value x ≥ 2),

18. WRITE Window\_Interval = (any new value that will result in an appropriate sample time),

19. VERIFY Attempted\_Samples = 0,

20. VERIFY Minimum\_Value = INF,

21. VERIFY Maximum\_Value = -INF,

22. VERIFY Average\_Value = NaN,

23. VERIFY Valid\_Samples = 0,

24. WAIT (at least two sample times),

25. VERIFY Minimum\_Value = (a value x: -INF < x < INF),

26. VERIFY Average\_Value = (a value ≠ NaN),

27. VERIFY Maximum\_Value = (a value x: Minimum\_Value ≤ x < INF),

28. VERIFY Attempted\_Samples = (a value x ≥ 2),

29. VERIFY Valid\_Samples = (a value x ≥ 2),

30. WRITE Window\_Samples = (any new value that will result in an appropriate sample time),

31. VERIFY Attempted\_Samples = 0,

32. VERIFY Minimum\_Value = INF,

33. VERIFY Maximum\_Value = -INF,

34. VERIFY Average\_Value = NaN,

35. VERIFY Valid\_Samples = 0,

36. IF (Object\_Property\_Reference is writable) THEN {

WAIT (at least two sample times),

VERIFY Minimum\_Value = (a value x: -INF < x < INF),

VERIFY Average\_Value = (a value ≠ NaN),

VERIFY Maximum\_Value = (a value x: Minimum\_Value ≤ x < INF),

VERIFY Attempted\_Samples = (a value x ≥ 2),

VERIFY Valid\_Samples = (a value x ≥ 2),

WRITE Object\_Property\_Reference = (any new value),

VERIFY Attempted\_Samples = 0,

VERIFY Minimum\_Value = INF,

VERIFY Maximum\_Value = -INF,

VERIFY Average\_Value = NaN,

VERIFY Valid\_Samples = 0

}

1. 管理样本窗口

BACnet参考条款：12.5.15

目的：验证Averaging对象是否正确跟踪样本中获得的平均值，最小值和最大值。这包括在采样窗口充满之前和之后进行监测。

测试思想：设置Averaging对象，监视可由测试代理或TD手动控制的属性。TD初始化样本，然后在每个采样间隔后监视Minimum\_Value，Average\_Value，Maximum\_Value，Attempted\_Samples和Valid\_Samples属性，验证其值是否正确跟踪监视的值。这需要能够操纵受监视属性值的值且要设置采样间隔使其时间足够慢以便进行分析。这将持续到样本窗口填满为止。如果IUT不支持此测试思想的Averaging对象配置，则应省略该测试。

配置要求：IUT应配置一个Averaging对象，用于监控可由测试代理或TD控制的属性。采样间隔应配置为允许时间更改受监视的属性值，并确定每个采样间隔后每个属性Minimum\_Value，Average\_Value，Maximum\_Value，Attempted\_Samples和Valid\_Samples是否正确更改。

测试步骤：

1. WRITE Attempted\_Samples = 0,

2. VERIFY Attempted\_Samples = 0,

3. VERIFY Minimum\_Value = INF,

4. VERIFY Maximum\_Value = -INF,

5. VERIFY Average\_Value = NaN,

6. VERIFY Valid\_Samples = 0,

7. REPEAT X = (1 to Window\_Samples + 5) DO {

WAIT (Window\_Interval / Window\_Samples)

IF (X ≤ Window\_Samples) THEN

VERIFY Attempted\_Samples = X

ELSE

VERIFY Attempted\_Samples = Window\_Samples,

VERIFY Minimum\_Value = (the minimum of the monitored values so far),

VERIFY Maximum\_Value = (the maximum of the monitored values so far),

VERIFY Average\_Value = (the average of the monitored values so far),

IF (X ≤ Window\_Samples) THEN

VERIFY Valid\_Samples = X

ELSE

VERIFY Valid\_Samples = Window\_Samples,

}

11. 二进制输入对象对象测试
12. 输入跟踪测试

所需测试：ReadProperty服务执行测试，10.18。

BACnet参考条款：12.6.4。

目的：验证跟踪和表示二进制输入值的能力。

配置要求：IUT连接到在测试期间可外部控制的二进制输入。

测试步骤：

1. MAKE (the real binary input ACTIVE)

2. VERIFY Present\_Value = ACTIVE

3. MAKE (the real binary input INACTIVE)

4. VERIFY Present\_Value = INACTIVE

1. Out\_Of\_Service，Status\_Flags和Reliability测试

验证Out\_Of\_Service属性的功能以及Status\_Flags，Reliability和Out\_Of\_Service之间关系的测试已在8.3.1.1中介绍。

1. 极性属性测试

所需测试：ReadProperty服务执行测试，10.18; WriteProperty服务执行测试，10.22。

BACnet参考条款：12.6.11。

目的：验证Polarity属性是否与关联的物理输入正确交互。如果Polarity属性不可写，则应省略该测试。

测试步骤：

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = (the Binary Input object being tested),

'Property Identifier' = Polarity

2. RECEIVE ReadProperty-ACK,

'Object Identifier' = (the Binary Input object being tested),

'Property Identifier' = Polarity

'Property Value' = NORMAL | REVERSE

3. TRANSMIT ReadProperty-Request,

'Object Identifier' = (the Binary Input object being tested),

'Property Identifier' = Present\_Value

4. RECEIVE ReadProperty-ACK,

'Object Identifier' = (the Binary Input object being tested),

'Property Identifier' = Present\_Value

'Property Value' = ACTIVE | INACTIVE

5. IF (the Polarity value in step 2 was NORMAL) THEN

WRITE Polarity = REVERSE

VERIFY Polarity = REVERSE

ELSE

WRITE Polarity = NORMAL

VERIFY Polarity = NORMAL

6. IF (the Present\_Value in step 4 was ACTIVE) THEN

VERIFY Present\_Value = INACTIVE

ELSE

VERIFY Present\_Value = ACTIVE

1. 状态属性测试变更

验证监视二进制输入对象状态变化能力的测试已在8.3.1.8中介绍。

1. 活动时间属性测试

验证监视二进制输入对象活动时间能力的测试已在8.3.1.9中介绍。

1. 内在报告测试

验证二进制输入对象内部报告功能的测试已在9.4.2中介绍。

11. 二进制输出对象测试
12. 输出跟踪测试

所需测试：ReadProperty服务执行测试，10.18; WriteProperty服务执行测试，10.22。

BACnet参考条款：12.7.4。

目的：验证表现和执行物理二进制输出的能力。

配置要求：IUT配置可在测试过程中进行观察的二进制输出。

测试步骤

1. WRITE Present\_Value = ACTIVE

2. VERIFY Present\_Value = ACTIVE

3. CHECK (verify that the output is active)

4. WRITE Present\_Value = INACTIVE

5. VERIFY Present\_Value = INACTIVE

6. CHECK (verify that the output is inactive)

1. Out\_Of\_Service，Status\_Flags和Reliability测试

验证Out\_Of\_Service属性的功能以及Status\_Flags，Reliability和Out\_Of\_Service之间关系的测试已在8.3.1.1中介绍。

1. 极性属性测试

所需测试：ReadProperty服务执行测试，10.18; WriteProperty服务执行测试，10.22。

BACnet参考条款：12.7.11。

目的：验证Polarity属性是否与关联的物理输出正确交互。如果Polarity属性不可写，则应省略该测试。

测试步骤：

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = (the Binary Output object being tested),

'Property Identifier' = Polarity

2. RECEIVE ReadProperty-ACK,

'Object Identifier' = (the Binary Output object being tested),

'Property Identifier' = Polarity,

'Property Value' = NORMAL | REVERSE

3. TRANSMIT ReadProperty-Request,

'Object Identifier' = (the Binary Output object being tested),

'Property Identifier' = Present\_Value

4. RECEIVE ReadProperty-ACK,

'Object Identifier' = (the Binary Output object being tested),

'Property Identifier' = Present\_Value,

'Property Value' = ACTIVE | INACTIVE

5. CHECK (the status of the physical output)

6. IF (the Polarity value in step 2 was NORMAL) THEN

WRITE Polarity = REVERSE

VERIFY Polarity = REVERSE

ELSE

WRITE Polarity = NORMAL

VERIFY Polarity = NORMAL

7. IF (the Present\_Value in step 4 was ACTIVE) THEN

VERIFY Present\_Value = ACTIVE

ELSE

VERIFY Present\_Value = INACTIVE

8. CHECK (the status of the physical output and verify that it is the complement of the status found in step 5)

1. 状态变更测试

验证监视二进制输出对象状态变化能力的测试已在8.3.1.8中介绍。

1. Elapsed\_Active\_Time属性测试

验证监视二进制输出对象活动时间能力的测试已在8.3.1.9中介绍。

1. 内在报告测试
2. 验证二进制输出对象的内在报告功能的测试见9.4.4.最小开启和最小关闭时间测试

验证最小开启和最小关闭时间算法操作的测试已在7.3.1.4和7.3.1.5中介绍。

1. 优先级命令测试

验证命令优先级排序算法操作的测试已在8.3.1.2和8.3.1.3中介绍。

11. 二进制值对象测试
12. Out\_Of\_Service，Status\_Flags和Reliability测试

验证Out\_Of\_Service属性的功能以及Status\_Flags，Reliability和Out\_Of\_Service之间关系的测试已在8.3.1.1中介绍。

1. 状态变更测试

验证监视二进制值对象状态变化能力的测试已在8.3.1.8中介绍。

1. Elapsed\_Active\_Time属性测试

验证监视二进制值对象活动时间能力的测试已在8.3.1.9中介绍。

1. 内在报告测试

验证二进制值对象的内部报告功能的测试将在9.4.2中介绍。

1. 最小开启和最小关闭时间测试

验证最小开启和最小关闭时间算法操作的测试已在8.3.1.4和8.3.1.5中介绍。

1. 优先级命令测试

验证命令优先级排序算法操作的测试已在8.3.1.2和8.3.1.3中介绍。

11. 日历测试

这些测试验证Calendar对象的Present\_Value属性是否与BACnet第12.9.6条定义的Date\_List具有关系。

1. 单日滚动测试

所需测试：ReadProperty 服务执行测试，10.18; TimeSynchronization服务执行测试，10.30。

BACnet参考条款：12.9。

目的：验证当Date\_List为单个日期形式时，表示日历状态的能力。必须支持执行TimeSynchronization服务，或者提供另一种方法来在测试期间重置IUT的时钟。

测试思想：Calendar对象配置有包含单个日期的Date\_List。IUT的时钟设置为紧接在Date\_List中指定的日期之前的日期和接近结束的时间。测试验证Calendar对象的Present\_Value最初是FALSE，并且当时间滚动到第二天时，Present\_Value变为TRUE。

配置要求：IUT应配置一个Calendar对象，该对象包含一个Date\_List，其中一个BACnetCalendarEntry以Date的形式出现。

测试步骤：

1. (TRANSMIT TimeSynchronization-Request,

'Time' = (the day preceding the one specified in Date\_List,

24:00:00 + UTC\_Offset – **Schedule Evaluation Fail Time** – 1 minute) ) |

MAKE (the local time = 24:00:00 – **Schedule Evaluation Fail Time** – 1 minute)

2. WAIT **Schedule Evaluation Fail Time**

3. VERIFY Present\_Value = FALSE

4. WAIT (**Schedule Evaluation Fail Time** + 2 minutes)

5. VERIFY Present\_Value = TRUE

1. 日期范围测试

所需测试：ReadProperty服务执行测试，10.18; TimeSynchronization服务执行测试，10.30。

BACnet参考条款：12.9。

目的：验证当Date\_List采用BACnetDateRange形式时，表示日历状态的能力。要么必须支持执行TimeSynchronization服务，要么必须提供其他方法在测试期间重置IUT的时钟。

测试思想：Calendar对象配置有包含单个BACnetDateRange的Date\_List。 IUT的时钟设置为日期范围之外的时间和日期。读取Present\_Value并验证为FALSE。时钟重置为日期范围内的值，并再次读取Present\_Value以验证其值为TRUE。如果IUT可以在日期范围内配置通配符字段，则应使用和不使用通配符进行测试。

配置要求：IUT应配置一个Calendar对象，该对象包含一个带有BACnetDateRange形式的单个BACnetCalendarEntry的Date\_List。

测试步骤：

1. (TRANSMIT TimeSynchronization-Request,

'Time' = (any day and time outside of the specified date range selected by the tester) ) |

MAKE (the local time = any day and time outside of the specified date range selected by the tester)

2. WAIT **Schedule Evaluation Fail Time**

3. VERIFY Present\_Value = FALSE

4. (TRANSMIT TimeSynchronization-Request,

'Time' = (any day and time inside the specified date range selected by the tester) ) |

MAKE (the local time = any day and time inside the specified date range selected by the tester)

5. WAIT **Schedule Evaluation Fail Time**

6. VERIFY Present\_Value = TRUE

1. WeekNDay测试

所需测试：ReadProperty服务执行测试，10.18; TimeSynchronization服务执行测试，10.30。

BACnet参考条款：12.9。

目的：验证当Date\_List采用BACnetWeekNDay形式时表示日历状态的能力。测试需要支持执行TimeSynchronization服务，或者必须提供其他方法来在测试期间重置IUT的时钟。

测试思想：Calendar对象配置有包含单个BACnetWeekNDay的Date\_List。IUT的时钟设置为与BACnetWeekNDay掩码匹配的时间和日期。读取Present\_Value并验证其为TRUE。时钟重置为与BACnetWeekNDay掩码匹配的值，月份除外。读取Present\_Value并验证为FALSE。时钟再次重置为与BACnetWeekNDay掩码匹配的值，除了该月的一周。读取Present\_Value并验证为FALSE。时钟再次重置为与BACnetWeekNDay掩码匹配的值，但星期几除外。读取Present\_Value并验证为FALSE。

配置要求：IUT应配置一个Calendar对象，该对象包含一个带有BACnetWeekNDay形式的单个BACnetCalendarEntry的Date\_List。

测试步骤：

1. (TRANSMIT TimeSynchronization-Request,

'Time' = (24-November-2001, 13:00:00 + UTC\_Offset) |

MAKE (the local time = 24-November-2001 at 13:00:00)

2. WAIT **Schedule Evaluation Fail Time**

3. VERIFY Present\_Value = TRUE

4. (TRANSMIT TimeSynchronization-Request,

'Time' = (27-October-2001, 13:00:00 + UTC\_Offset) |

MAKE (the local time = 27-October-2001 at 13:00:00)

5. WAIT **Schedule Evaluation Fail Time**

6. VERIFY Present\_Value = FALSE

7. (TRANSMIT TimeSynchronization-Request,

'Time' = (17-November-2001, 13:00:00 + UTC\_Offset) |

MAKE (the local time = 17-November-2001 at 13:00:00)

8. WAIT **Schedule Evaluation Fail Time**

9. VERIFY Present\_Value = FALSE

10. (TRANSMIT TimeSynchronization-Request,

'Time' = (25-November-2001, 13:00:00 + UTC\_Offset) |

MAKE (the local time = 25-November-2001 at 13:00:00)

11. WAIT **Schedule Evaluation Fail Time**

12. VERIFY Present\_Value = FALSE

11. 命令对象测试
12. 所有写入成功后延迟测试

所需测试：ReadProperty服务执行，10.18; WriteProperty服务执行测试，10.22。

BACnet参考条款：12.10.8。

目的：验证Command对象是否可以成功执行包含发布延迟的操作列表。

测试思想：IUT配置一个动作列表，其中包括操作一系列外部可见输出，每个输出之间有一个时间延迟。TD触发此动作列表，测试人员观察外部变化。如果IUT不支持后延迟，则应省略该测试。如果IUT不支持此测试思想的动作列表配置，也省略该测试。

配置要求：IUT应配置一个具有动作列表X的Command对象，其中包括写一系列外部可见输出。写入外部可见输出之间应有足够大的后延迟，以使测试人员观察到延迟。

测试步骤：

1. WRITE Present\_Value = X

2. RECEIVE Simple-ACK-PDU

3. VERIFY In\_Process = TRUE

4. CHECK (for the externally visible actions and verify that there is a post delay)

5. VERIFY In\_Process = FALSE

6. VERIFY All\_Writes\_Successful = TRUE

1. 故障退出测试

所需测试：ReadProperty服务执行测试，10.18; WriteProperty服务执行测试，10.22。

BACnet参考条款：12.10.8。

目的：验证Command对象是否可以成功执行Quit\_On\_Failure过程。

测试思想：IUT配置有两个动作列表，其中包括一系列外部可见输出，并且序列中的某个位置将会失败。两个动作列表除了一个将Quit\_On\_Failure设置为TRUE而另一个设置为FALSE之外，其余部分是相同的。TD触发两个动作列表。观察外部输出以验证故障程序被执行。如果IUT不支持此测试思想的动作列表配置，则应省略该测试。

配置要求：IUT应配置一个Command对象，该对象至少有两个动作列表X和Y，动作列表包括写入一系列外部可见输出。序列中的某个地方应该有一个将会失败的写入命令，其后跟踪一个会成功的写入命令。除了列表X将Quit\_On\_Failure设置为TRUE且Y应将Quit\_On\_Failure设置为FALSE之外，两个动作列表都应相同。

测试步骤：

1. WRITE Present\_Value = X

2. RECEIVE Simple-ACK-PDU

3. VERIFY In\_Process = TRUE

4. CHECK (for the externally visible actions and verify that they stop when the failure occurs)

5. VERIFY In\_Process = FALSE,

6. VERIFY All\_Writes\_Successful = FALSE

7. WRITE Present\_Value = Y

8. RECEIVE Simple-ACK-PDU

9. VERIFY In\_Process = TRUE

10. CHECK (for the externally visible actions and verify that they continue to the end after the failure occurs)

11. VERIFY In\_Process = FALSE,

12. VERIFY All\_Writes\_Successful = FALSE

1. 外部写入测试

所需测试：ReadProperty服务执行测试，10.18; WriteProperty服务执行测试，10.22。

BACnet参考条款：12.10.8。

目的：验证Command对象是否可以写入外部对象。 如果IUT不支持从Command对象写入外部对象，则应省略该测试。

测试思想：IUT配置有Command对象，该Command对象具有包括写入TD中的对象的动作列表。 TD通过将适当的值写入Command对象来调用此操作列表。 TD验证IUT发送适当的WriteProperty-Request。

配置要求：IUT应配置一个Command对象，该对象具有一个Action属性，该属性包含一个动作列表X，其中包含一个写入TD中（Analog Value，0）的Present\_Value的命令。

测试步骤：

1. WRITE Present\_Value = X

2. RECEIVE Simple-ACK-PDU

3. BEFORE **External Command Fail Time**

RECEIVE WriteProperty-Request,

'Object Identifier' = (Analog Value, 0),

'Property Identifier' = Present\_Value,

'Property Value' = (any Real value)

4. VERIFY In\_Process = TRUE

5. TRANSMIT BACnet-Simple-ACK

6. VERIFY In\_Process = FALSE

注意事项：IUT生成的任何WriteProperty请求都可能具有Priority参数。如果包括在内，则应在1-16范围内，不包括6。

1. 空动作列表测试

所需测试：WriteProperty服务执行测试，10.22。

BACnet参考条款：12.10.8。

目的：验证使用与空操作列表对应的非零值写入Present\_Value时，Command对象不执行任何操作。

测试思想：IUT配置有至少一个空动作列表。TD触发动作列表，观察外部输出验证没有发生变化。如果IUT不支持此测试思想的动作列表配置，则应省略该测试。

配置要求：IUT应配置一个Command对象，该对象具有至少有一个空操作列表的Action属性。

测试步骤：

1. WRITE Present\_Value = (an index corresponding to an empty action list)

2. RECEIVE Simple-ACK-PDU

3. VERIFY In\_Process = FALSE

4. VERIFY All\_Writes\_Successful = TRUE

5. CHECK (if any of the actions of the Command object have externally visible results verify that no changes occurred)

1. Action 0测试

所需测试：WriteProperty服务执行测试，10.22。

BACnet参考条款：12.10.8。

目的：验证Present\_Value写入零值时，命令对象不执行任何操作。

配置要求：IUT应配置有一个Command对象，该Command对象具有一个Action属性，该属性具有至少一个非空的动作列表。

测试步骤：

1. WRITE Present\_Value = 0

2. RECEIVE Simple-ACK-PDU

3. VERIFY In\_Process = FALSE

4. VERIFY All\_Writes\_Successful = TRUE

5. CHECK (if any of the actions of the Command object have externally visible results verify that no changes occurred)

1. Action\_Text测试

所需测试：WriteProperty服务执行测试，10.22。

BACnet参考条款：12.10.8和12.10.9。

目的：验证Action数组的大小是否与Action\_Text数组的大小相对应。

测试步骤：

1. TRANSMIT ReadProperty-Request,

‘Object Identifier’ = (the Command object being tested),

‘Property Identifier’ = Action,

‘Property Array Index’ = 0

2. RECEIVE ReadProperty-ACK,

‘Object Identifier’ = (the Command object being tested),

‘Property Identifier’ = Action,

‘Property Value’ = (any integer greater than 0)

3. VERIFY Action\_Text = (the size of the Action array from step 2), ARRAY INDEX = 0

1. In\_Process为TRUE时，写测试

所需测试：WriteProperty服务执行测试，10.22。

BACnet参考条款：12.10.8和12.10.9。

目的：在In\_Process为TRUE的情况下，命令第二个动作列表，并且未执行第二个动作列表，验证动作列表是否继续完成。

测试方案：IUT配置有两个含外部可见输出的Action列表，每个Action都有后延迟。TD触发第一个Action列表，观察外部输出，以便在第一个列表的后延迟期间触发第二个Action列表。TD触发第二个Action列表，观察外部输出，以验证是否执行第二个Action列表。如果IUT不支持后延迟，则应省略该测试。如果IUT不支持该测试概念的Action列表配置，也省略该测试。

配置要求：IUT配置一个命令对象，该对象具有两个不同的Action列表X和Y，可写入外部可见输出。外部可见输出之间应该有一个测试人员可观测到的后延迟（这可确保In\_Process保持TRUE足够长的时间来命令第二个Action列表）。

测试步骤：

1. WRITE Present\_Value = X

2. RECEIVE Simple-ACK-PDU

3. WRITE Present\_Value = Y

4. RECEIVE BACnet-Error-PDU

Error Class = SERVICES,

Error Code = SERVICE\_REQUEST\_DENIED | OTHER

5. CHECK (that the externally visible actions of X took place)

6. CHECK (that the externally visible actions of Y did not take place)

7. VERIFY In\_Process = FALSE,

8. VERIFY All\_Writes\_Successful = TRUE

1. Action大小改变Action\_Text大小测试

所需测试：WriteProperty服务执行测试，10.22。

BACnet参考条款：12.10.8和12.10.9。

目的：验证当更改Action数组的大小时，Action\_Text数组的大小也相应更改为相同的大小。如果不能更改Action和Action\_Text数组的大小，则不执行此测试。如果Protocol\_Revision不存在或值小于4，也不执行此测试。

配置要求：IUT应配置有可调整大小的Action和Action\_Text数组的Command对象。

测试方案：Action和Action\_Text数组设置为特定值。通过写入Action\_Text数组元素0来增大，通过写入Action\_Text数组来减小；再通过写入Action\_Text数组来增大，通过写入Action\_Text数组元素0来减小。

测试步骤：

1. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Command object being tested),

‘Property Identifier’ = Action,

‘Property Array Index’ = 0,

‘Property Value’ = 2

2. RECEIVE Simple-ACK-PDU

3. VERIFY Action = 2, ARRAY INDEX = 0

4. VERIFY Action\_Text = 2, ARRAY INDEX = 0

5. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Command object being tested),

‘Property Identifier’ = Action,

‘Property Array Index’ = 0,

‘Property Value’ = (some value greater than 2)

6. RECEIVE Simple-ACK-PDU

7. VERIFY Action = (the value written in step 5), ARRAY INDEX = 0

8. VERIFY Action\_Text = (the value written in step 5), ARRAY INDEX = 0

9. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Command object being tested),

‘Property Identifier’ = Action,

‘Property Value’ = (Action array of length 2)

10. RECEIVE Simple-ACK-PDU

11. VERIFY Action = 2, ARRAY INDEX = 0

12. VERIFY Action\_Text = 2, ARRAY INDEX = 0

13. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Command object being tested),

‘Property Identifier’ = Action,

‘Property Value’ = (Action array of length greater than 2)

14. RECEIVE Simple-ACK-PDU

15. VERIFY Action = (the length of the array written in step 13), ARRAY INDEX = 0

16. VERIFY Action\_Text = (the length of the array written in step 13), ARRAY INDEX = 0

17. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Command object being tested),

‘Property Identifier’ = Action,

‘Property Array Index’ = 0,

‘Property Value’ = 2

18. RECEIVE Simple-ACK-PDU

19. VERIFY Action = (an array consisting of elements 1 & 2 from the array written in step 13)

20. VERIFY Action\_Text = 2, ARRAY INDEX = 0

1. Action\_Text大小改变动作大小测试

所需测试：WriteProperty服务执行测试，10.22

BACnet参考条款：12.10.8和12.10.9

目的：验证当Action\_Text数组的大小发生更改时，Action数组的大小会相应更改为相同值。如果无法更改Action和Action\_Text数组的大小，则不执行此测试。

配置要求：IUT应配置有可调整大小的Action和Action\_Text数组的Command对象。

测试方案：Action和Action\_Text数组设置为特定值。通过写入Action\_Text数组元素0增大，通过写入Action\_Text数组来减小；再通过写入Action\_Text数组来增大，通过写入Action\_Text数组元素0来减小。

测试步骤：

1. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Command object being tested),

‘Property Identifier’ = Action\_Text,

‘Property Array Index’ = 0,

‘Property Value’ = 2

2. RECEIVE Simple-ACK-PDU

3. VERIFY Action\_Text = 2, ARRAY INDEX = 0

4. VERIFY Action = 2, ARRAY INDEX = 0

5. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Command object being tested),

‘Property Identifier’ = Action\_Text,

‘Property Array Index’ = 0,

‘Property Value’ = (some value greater than 2)

6. RECEIVE Simple-ACK-PDU

7. VERIFY Action\_Text = (the value written in step 5), ARRAY INDEX = 0

8. VERIFY Action = (the value written in step 5), ARRAY INDEX = 0

9. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Command object being tested),

‘Property Identifier’ = Action\_Text,

‘Property Value’ = (Action\_Text array of length 2)

10. RECEIVE Simple-ACK-PDU

11. VERIFY Action\_Text = 2, ARRAY INDEX = 0

12. VERIFY Action = 2, ARRAY INDEX = 0

13. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Command object being tested),

‘Property Identifier’ = Action\_Text,

‘Property Value’ = (Action\_Text array of length greater than 2)

14. RECEIVE Simple-ACK-PDU

15. VERIFY Action\_Text = (the length of the array written in step 13), ARRAY INDEX = 0

16. VERIFY Action = (the length of the array written in step 13), ARRAY INDEX = 0

17. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Command object being tested),

‘Property Identifier’ = Action\_Text,

‘Property Array Index’ = 0,

‘Property Value’ = 2

18. RECEIVE Simple-ACK-PDU

19. VERIFY Action\_Text = (an array consisting of elements 1 & 2 from the array written in step 13)

20. VERIFY Action = 2, ARRAY INDEX = 0

11. 设备对象测试

设备对象功能的大多数必要测试都包含在对应用程序服务或它们所对应的特殊功能的测试中。

1. Active\_COV\_Subscriptions SubscribeCOV测试

目的：此测试用例验证在使用SubscribeCOV创建、取消和超时COV订阅时，IUT是否正确更新了Active\_COV\_Subscriptions属性。

配置要求：在此测试中，测试人员应选择支持SubscribeCOV的三个标准对象O1,O2,O3。O1,O2,O3不需要适用不同对象。测试人员还应选择三个非零的唯一过程标识符P1,P2和P3以及三个非零的生存期L1,L2和L3。L1长度应足够长，以允许测试从开始运行到步骤14，L2和L3长度应足够长，以支持完成整个测试。

IUT开始测试时，其Active\_COV\_Subscriptions属性中没有任何条目。

测试步骤：

1. TRANSMIT SubscribeCOV-Request,

‘Subscriber Process Identifier’ = P1,

‘Monitored Object Identifier’ = O1,

‘Issue Confirmed Notifications’ = TRUE,

‘Lifetime’ = L1

2. RECEIVE BACnet-SimpleACK-PDU

3. BEFORE **NotificationFailTime**

RECEIVE ConfirmedCOVNotification-Request,

‘Subscriber Process Identifier’ = P1,

‘Initiating Device Identifier’ = IUT,

‘Monitored Object Identifier’ = O1,

‘Time Remaining’ = (a value approximately equal to L1),

‘List of Values’ = (values appropriate to the object type of the monitored object)

4. TRANSMIT BACnet-SimpleACK-PDU

5. VERIFY Active\_COV\_Subscriptions = {{ {TD, P1}, {O1, Present\_Value }, TRUE, (a value less than L1), (a valid Increment if the property is REAL) }}

6. TRANSMIT SubscribeCOV-Request,

‘Subscriber Process Identifier’ = P2,

‘Monitored Object Identifier’ = O2,

‘Issue Confirmed Notifications’ = FALSE,

‘Lifetime’ = L2

7. RECEIVE BACnet-SimpleACK-PDU

8. BEFORE **NotificationFailTime**

RECEIVE UnconfirmedCOVNotification-Request,

‘Subscriber Process Identifier’ = P2,

‘Initiating Device Identifier’ = IUT,

‘Monitored Object Identifier’ = O2,

‘Time Remaining’ = (a value approximately equal to L2),

‘List of Values’ = (values appropriate to the object type of the monitored object)

9. VERIFY Active\_COV\_Subscriptions = {{ {TD, P1}, {O1, Present\_Value}, TRUE, (a value less than L1), (a valid Increment if the property is REAL) },

{ {TD, P2}, {O2, Present\_Value}, FALSE, (a value less than L2), (a valid Increment if the property is REAL) }}

10. TRANSMIT SubscribeCOV-Request,

‘Subscriber Process Identifier’ = P3,

‘Monitored Object Identifier’ = O3,

‘Issue Confirmed Notifications’ = FALSE,

‘Lifetime’ = L3

11. RECEIVE BACnet-SimpleACK-PDU

12. BEFORE **NotificationFailTime**

RECEIVE UnconfirmedCOVNotification-Request,

‘Subscriber Process Identifier’ = P3,

‘Initiating Device Identifier’ = IUT,

‘Monitored Object Identifier’ = O3,

‘Time Remaining’ = (a value approximately equal to L3),

‘List of Values’ = (values appropriate to the object type of the monitored object)

13. VERIFY Active\_COV\_Subscriptions = {{{TD, P1}, {O1, Present\_Value}, TRUE, (a value less than L1), (a valid Increment if the property is REAL)},

{{TD, P2}, {O2, Present\_Value}, FALSE, (a value less than L2), (a valid Increment if the property is REAL)},

{{TD, P3}, {O3, Present\_Value}, FALSE, (a value less than L3), (a valid Increment if the property is REAL)}}

14. WAIT L1 + the IUT’s timer granularity

1. Active\_COV\_Subscriptions SubscribeCOVProperty测试

目的：此测试用例验证在使用SubscribeCOV属性创建、取消和超时COV订阅时，IUT是否正确更新了Active\_COV\_Subscriptions属性。

配置要求：在此测试中，测试人员应选择支持SubscribeCOV属性的三个标准对象O1,O2,O3。O1,O2,O3不需要适用不同对象。测试人员还应选择三个非零的唯一过程标识符P1,P2和P3以及三个非零的生存期L1,L2和L3。L1长度应足够长，以允许测试从开始运行到步骤14，L2和L3长度应足够长，以支持完成整个测试。

测试步骤：与第7.3.2.10.1节中的测试步骤相同，除了使用SubscribeCOVProperty而不是SubscribeCOV，并且Active\_COV\_Subscriptions应反映实际订阅属性而不是Present\_Value（如果订阅属性不是Present\_Value）。

1. 创建对象后Database\_Revision属性成功增加

所需测试：ReadProperty服务执行测试，10.18

BACnet参考条款：12.11.34

目的：验证创建对象时Device对象的Database\_Revision属性是否递增。若无法创建对象，则省略该测试。

测试方案：读取Device对象的Database\_Revision属性。创建一个对象，再次读取Device对象的Database\_Revision属性以验证它是否递增。

配置要求：无。

测试步骤：

1. TRANSMIT ReadProperty-Request,

‘Object Identifier’ = (the Device object),

‘Property Identifier’ = Database\_Revision

2. RECEIVE ReadProperty-ACK,

‘Object Identifier’ = (the Device object),

‘Property Identifier’ = Database\_Revision,

‘Property Value’ = (any value = initial value)

3. MAKE (create an object)

4. TRANSMIT ReadProperty-Request,

‘Object Identifier’ = (the Device object),

‘Property Identifier’ = Database\_Revision

5. RECEIVE ReadProperty-ACK,

‘Object Identifier’ = (the Device object),

‘Property Identifier’ = Database\_Revision,

‘Property Value’ = (greater than initial value taking wrapping into account)

1. 删除对象后Database\_Revision属性成功增加

所需测试：ReadProperty服务执行测试，10.18

BACnet参考条款：12.11.34

目的：验证删除对象时Device对象的Database\_Revision属性是否递增。若无法创建对象，则省略该测试。

测试方案：读取Device对象的Database\_Revision属性。删除一个对象，再次读取Device对象的Database\_Revision属性以验证它是否递增。

配置要求：无。

测试步骤：

1. TRANSMIT ReadProperty-Request,

‘Object Identifier’ = (the Device object),

‘Property Identifier’ = Database\_Revision

2. RECEIVE ReadProperty-ACK,

‘Object Identifier’ = (the Device object),

‘Property Identifier’ = Database\_Revision,

‘Property Value’ = (any value = initial value)

3. MAKE (delete an object)

4. TRANSMIT ReadProperty-Request,

‘Object Identifier’ = (the Device object),

‘Property Identifier’ = Database\_Revision

5. RECEIVE ReadProperty-ACK,

‘Object Identifier’ = (the Device object),

‘Property Identifier’ = Database\_Revision,

‘Property Value’ = (greater than initial value taking wrapping into account)

1. 更改Object\_Name属性后Database\_Revision属性成功增加

所需测试：ReadProperty服务执行测试，10.18

BACnet参考条款：12.11.34

目的：验证更改Object\_Name属性时设备对象的Database\_Revision属性是否递增。若无法创建对象，则省略该测试。

测试方案：读取Device对象的Database\_Revision属性。更改对象的Object\_Name属性，再次读取Device对象的Database\_Revision属性以验证它是否递增。

配置要求：无。

测试步骤：

1. TRANSMIT ReadProperty-Request,

‘Object Identifier’ = (the Device object),

‘Property Identifier’ = Database\_Revision

2. RECEIVE ReadProperty-ACK,

‘Object Identifier’ = (the Device object),

‘Property Identifier’ = Database\_Revision,

‘Property Value’ = (any value = initial value)

3. MAKE (the Object\_Name property of an object change)

4. TRANSMIT ReadProperty-Request,

‘Object Identifier’ = (the Device object),

‘Property Identifier’ = Database\_Revision

5. RECEIVE ReadProperty-ACK,

‘Object Identifier’ = (the Device object),

‘Property Identifier’ = Database\_Revision,

‘Property Value’ = (greater than initial value taking wrapping into account)

1. 更改Object\_Identifier属性后Database\_Revision属性成功增加

所需测试：ReadProperty服务执行测试，10.18

BACnet参考条款：12.11.34

目的：验证更改Object\_Identifier属性时Device对象的Database\_Revision属性是否递增。若无法创建对象，则省略该测试。

测试方案：读取Device对象的Database\_Revision属性。更改对象名，再次读取Device对象的Database\_Revision属性，验证它是否递增。

配置要求：无。

测试步骤：

1. TRANSMIT ReadProperty-Request,

‘Object Identifier’ = (the Device object),

‘Property Identifier’ = Database\_Revision

2. RECEIVE ReadProperty-ACK,

‘Object Identifier’ = (the Device object),

‘Property Identifier’ = Database\_Revision,

‘Property Value’ = (any value = initial value)

3. MAKE (the Object\_Identifier property of an object change)

4. TRANSMIT ReadProperty-Request,

‘Object Identifier’ = (the Device object),

‘Property Identifier’ = Database\_Revision

5. RECEIVE ReadProperty-ACK,

‘Object Identifier’ = (the Device object),

‘Property Identifier’ = Database\_Revision,

‘Property Value’ = (greater than initial value taking wrapping into account)

11. 事件注册对象测试

除了本节中的测试之外，用于验证事件注册Event Enrollment对象功能的其他测试将在9.4和9.5中启动事件报告测试中介绍。如果IUT支持IUT之外的监视对象，则应使用9.4和9.5中的一项测试来证明该能力。这需要另外提供一个配置正确的BACnet设备。

1. Event\_Type测试

所需测试：ReadProperty服务执行测试，10.18;WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.5。

目的：验证Event Enrollment对象的Event\_Type属性是否正确跟踪Event\_Parameters属性更改。

测试方案：写入Event\_Parameters属性，验证Event\_Type的更改。

配置要求：IUT配置一个Event Enrollment对象E1，E1具有可写的Event\_Parameters属性，该属性将写入由不同算法选择的值。测试人员应选择一个有效的Event\_Parameters值EP1，其Event\_Type值为ET1，IUT将写入ET1与E1中已存在的Event\_Type不同的值。

1. VERIFY Event\_Type <> ET1

2. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = E1,

‘Property Identifier’ = Event\_Parameters,

‘Property Value’ = EP1

3. RECEIVE BACnet-SimpleACK-PDU

4. VERIFY Event\_Type = ET1

11. File对象测试

File对象测试将在10.12和10.13执行文件访问服务测试中介绍。

11. GlobalGroup对象测试
12. 写入Group\_Members属性调整Group\_Member\_Names大小测试

所需测试：WriteProperty服务执行测试，10.22

目的：验证当通过写入更改Group\_Members数组的大小时，Group\_Member\_Names和Present\_Value数组的大小会相应更改，并包含指定的初始化值。如果无法写入Group\_Members数组，则不执行此测试。

配置要求：IUT应配置一个具有Group\_Members属性可写的Global Group对象。

测试方案：Group\_Members数组设置为特定大小。然后通过写入数组大小来增大，通过写入数组来减小，通过写入数组来增加，通过写入数组大小来减少。每个步骤都验证Group\_Member\_Names和Present\_Value数组的大小，并检查新元素的初始化值（如果有的话）。

测试步骤：

1. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Global Group object being tested),

‘Property Identifier’ = Group\_Members,

‘Property Array Index’ = 0,

‘Property Value’ = 2

2. RECEIVE Simple-ACK-PDU

3. VERIFY Group\_Members = 2, ARRAY INDEX = 0

4. VERIFY Group\_Member\_Names = 2, ARRAY INDEX = 0

5. VERIFY Present\_Value = 2, ARRAY INDEX = 0

6. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Global Group object being tested),

‘Property Identifier’ = Group\_Members,

‘Property Array Index’ = 0,

‘Property Value’ = (some value greater than 2)

7. RECEIVE Simple-ACK-PDU

8. VERIFY Group\_Members = (the value written in step 6), ARRAY INDEX = 0

9. VERIFY Group\_Member\_Names = (the value written in step 6), ARRAY INDEX = 0

10. VERIFY Present\_Value = (the value written in step 6), ARRAY INDEX = 0

11. VERIFY Group\_Members = (a BACnetDeviceObjectPropertyReference containing

(Device, Instance number 4194303)), ARRAY INDEX = (some value from 3 through the value written in step 6)

12. VERIFY Group\_Member\_Names = (an empty string),

ARRAY INDEX = (some value from 3 through the value written in step 6)

13. VERIFY Present\_Value = ‘Access\_Result’ = PropertyAccessError (PROPERTY, VALUE\_NOT\_INITIALIZED),

ARRAY INDEX = (some value from 3 through the value written in step 6)

14. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Global Group object being tested),

‘Property Identifier’ = Group\_Members,

‘Property Value’ = (a one-element array containing any valid

BACnetDeviceObjectPropertyReference)

15. RECEIVE Simple-ACK-PDU

16. VERIFY Group\_Members = 1, ARRAY INDEX = 0

17. VERIFY Group\_Member\_Names = 1, ARRAY INDEX = 0

18. VERIFY Present\_Value = 1, ARRAY INDEX = 0

19. VERIFY Group\_Members = (the array written in step 14)

20. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Global Group object being tested),

‘Property Identifier’ = Group\_Members,

‘Property Value’ = (an array of two or more valid

BACnetDeviceObjectPropertyReference values)

21. RECEIVE Simple-ACK-PDU

22. VERIFY Group\_Members = (the size of the array written in step 20), ARRAY INDEX = 0

23. VERIFY Group\_Member\_Names = (the size of the array written in step 20), ARRAY INDEX = 0

24. VERIFY Present\_Value = (the size of the array written in step 20), ARRAY INDEX = 0

25. VERIFY Group\_Members = (the array written in step 20)

26. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Global Group object being tested),

‘Property Identifier’ = Group\_Members,

‘Property Array Index’ = 0,

‘Property Value’ = (some value between 0 and the size of the array written in step 20)

27. RECEIVE Simple-ACK-PDU

28. VERIFY Group\_Members = (the size of the array written in step 26), ARRAY INDEX = 0

29. VERIFY Group\_Member\_Names = (the size of the array written in step 26), ARRAY INDEX = 0

30. VERIFY Present\_Value = (the size of the array written in step 26), ARRAY INDEX = 0

1. Group 写入Group\_Member\_Names属性调整Group\_Member大小测试测试

所需测试：WriteProperty服务执行测试，10.22

目的：验证当通过写入更改Group\_Member\_Names数组的大小时，Group\_Members和Present\_Value数组的大小会相应更改，并且任何新条目都包含指定的初始化值。如果无法写入Group\_Member\_Names数组，则不执行此测试。

配置要求：IUT应配置一个具有可写Group\_Member\_Names属性的Global Group对象。

测试方案：Group\_Member\_Names数组设置为特定大小。通过写入数组大小来增大，通过写入数组减小，通过写入数组增加，通过写入数组大小减小。每个步骤都验证Group\_Member和Present\_Value数组的大小，并检查新元素的初始化值。

测试步骤：

1. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Global Group object being tested),

‘Property Identifier’ = Group\_Member\_Names,

‘Property Array Index’ = 0,

‘Property Value’ = 2

2. RECEIVE Simple-ACK-PDU

3. VERIFY Group\_Member\_Names = 2, ARRAY INDEX = 0

4. VERIFY Group\_Members = 2, ARRAY INDEX = 0

5. VERIFY Present\_Value = 2, ARRAY INDEX = 0

6. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Global Group object being tested),

‘Property Identifier’ = Group\_Member\_Names,

‘Property Array Index’ = 0,

‘Property Value’ = (some value greater than 2)

7. RECEIVE Simple-ACK-PDU

8. VERIFY Group\_Member\_Names = (the value written in step 6), ARRAY INDEX = 0

9. VERIFY Group\_Members = (the value written in step 6), ARRAY INDEX = 0

10. VERIFY Present\_Value = (the value written in step 6), ARRAY INDEX = 0

11. VERIFY Group\_Member\_Names = (an empty string),

ARRAY INDEX = (some value from 3 through the value written in step 6)

12. VERIFY Group\_Members = (Device, Instance number 4194303),

ARRAY INDEX = (some value from 3 through the value written in step 6)

13. VERIFY Present\_Value = ‘Access\_Result’ = PropertyAccessError (PROPERTY, VALUE\_NOT\_INITIALIZED),

ARRAY INDEX = (some value from 3 through the value written in step 6)

14. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Global Group object being tested),

‘Property Identifier’ = Group\_Member\_Names,

‘Property Value’ = (an array of one Character String)

15. RECEIVE Simple-ACK-PDU

16. VERIFY Group\_Member\_Names = 1, ARRAY INDEX = 0

17. VERIFY Group\_Members = 1, ARRAY INDEX = 0

18. VERIFY Present\_Value = 1, ARRAY INDEX = 0

19. VERIFY Group\_Member\_Names = (the array written in step 14)

20. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Global Group object being tested),

‘Property Identifier’ = Group\_Member\_Names,

‘Property Value’ = (an array of two or more Character Strings)

21. RECEIVE Simple-ACK-PDU

22. VERIFY Group\_Member\_Names = (the size of the array written in step 20), ARRAY INDEX = 0

23. VERIFY Group\_Members = (the size of the array written in step 20), ARRAY INDEX = 0

24. VERIFY Present\_Value = (the size of the array written in step 20), ARRAY INDEX = 0

25. VERIFY Group\_Member\_Names = (the array of Character Strings written in step 20)

26. TRANSMIT WriteProperty-Request, ‘Object Identifier’ = (the Global Group object being tested),

‘Property Identifier’ = Group\_Member\_Names,

‘Property Array Index’ = 0,

‘Property Value’ = (some value between 0 and the size of the array written in step 20)

27. RECEIVE Simple-ACK-PDU

28. VERIFY Group\_Member\_Names = (the size of the array written in step 26), ARRAY INDEX = 0

29. VERIFY Group\_Members = (the size of the array written in step 26), ARRAY INDEX = 0

30. VERIFY Present\_Value = (the size of the array written in step 26), ARRAY INDEX = 0

1. Group 对象测试

所需测试：ReadProperty服务执行测试，10.18; WriteProperty服务执行测试，10.22。

BACnet参考条款：12.14。

目的：验证一组Present\_Value属性可正确跟踪构成该组的对象的属性值变化。

测试方案：读取一个Group对象的Present\_Value属性和构成该Group的每个成员的对象和属性，比较这些值，验证是否匹配。更改其中的一个Group成员的值，再次读取该Group的Present\_Value属性，验证是否正确跟踪更改。

配置要求：IUT配置一个至少有两个成员的Group对象，其中一个组成员可以通过WriteProperty服务或供应商提供的其他方式进行更改。除了测试程序需要更改的部分外，构成本Group的属性值在测试期间应保持不变。

测试步骤：

1. TRANSMIT ReadProperty-Request,

‘Object Identifier’ = (the Group object being tested),

‘Property Identifier’ = List\_Of\_Group\_Members

2. RECEIVE ReadProperty-ACK,

‘Object Identifier’ = (the Group object being tested),

‘Property Identifier’ = List\_Of\_Group\_Members,

‘Property Value’ = (any valid list of group members)

3. TRANSMIT ReadProperty-Request,

‘Object Identifier’ = (the Group object being tested),

‘Property Identifier’ = Present\_Value

4. RECEIVE ReadProperty-ACK,

‘Object Identifier’ = (the Group object being tested),

‘Property Identifier’ = List\_Of\_Group\_Members,

‘Property Value’ = (any valid set of values consistent with the properties that make up the group)

5. REPEAT X = (each object, property combination returned in the List\_Of\_Group\_Members in step 2) DO {

VERIFY X = (the same value that was returned for this group member in step 4)}

6. IF (a property value of a group member is changeable) THEN

IF (the changeable group member property value is writable) THEN

WRITE (the writable property that is a member of the group) = (a value different from its current value)

ELSE

MAKE (the changeable group member property value different from its current value)

7. WAIT **Internal Processing Fail Time**

8. TRANSMIT ReadProperty-Request,

‘Object Identifier’ = (the Group object being tested),

‘Property Identifier’ = Present\_Value

9. RECEIVE ReadProperty-ACK,

‘Object Identifier’ = (the Group object being tested),

‘Property Identifier’ = List\_Of\_Group\_Members,

‘Property Value’ = (the same set of values received in step 4 except for the value changed in step 6)

10. REPEAT X = (each object, property combination returned in the List\_Of\_Group\_Members in step 2) DO {

VERIFY X = (the same value that was returned for this group member in step 9)}

1. 生命安全点对象测试
2. Tracking Value测试

所需测试：ReadProperty服务执行测试，10.18。

BACnet参考条款：12.15.5。

目的：验证在对象处于NORMAL状态时，生命安全点对象的Present\_Value和Tracking Value属性跟踪其改变。如果不支持可选Tracking Value属性，则省略此测试。

测试步骤：

1. MAKE (Present\_Value have a value x such that x corresponds to a NORMAL state)

2. VERIFY Tracking Value = Present Value

3. MAKE (Present\_Value have a value x such that x corresponds to a NORMAL state but different from the previous value)

4. VERIFY Tracking Value = Present Value

1. Out\_Of\_Service、Status\_Flags和Reliability测试

验证Out\_Of\_Service属性的功能以及Out\_Of\_Service、Status\_Flags和Reliability之间的关系的测试已在8.3.1.1中介绍。

1. 内部报告测试

验证生命安全点对象的内部报告功能的测试将在9.4.8中介绍。

1. 模式测试

验证Mode属性更改时的生命安全点对象的内部报告功能的测试将在9.4.8中介绍。

11. 生命安全区对象测试
12. Tracking Value测试

所需测试：ReadProperty服务执行测试，10.18。

BACnet参考条款：12.16.5。

目的：验证在对象处于NORMAL状态时，生命安全区对象的Present\_Value和Tracking\_Value属性跟踪其改变。如果不支持可选Tracking Value属性，则省略此测试。

测试步骤：

1. MAKE (Present\_Value have a value x such that x corresponds to a NORMAL state)

2. VERIFY Tracking Value = Present Value

3. MAKE (Present\_Value have a value x such that x corresponds to a NORMAL state but different from the previous value)

4. VERIFY Tracking Value = Present Value

1. Out\_Of\_Service、Status\_Flags和Reliability测试

验证Out\_Of\_Service属性的功能以及Out\_Of\_Service、Status\_Flags和Reliability之间的关系的测试已在8.3.1.1中介绍。

1. 内部报告测试

验证生命安全点对象的内部报告功能的测试将在9.4.8中介绍。

1. 模式测试

验证Mode属性更改时的生命安全点对象的内部报告功能的测试将在9.4.8中介绍。

11. 循环对象测试
12. Manipulated\_Variable\_Reference 跟踪

所需测试：ReadProperty服务执行测试，10.18。

BACnet参考条款：12.17.12。

目的：验证Manipulated\_Variable\_Reference属性是否跟踪循环对象的Present\_Value改变。

配置要求：配置IUT，使测试循环Loop对象的控制输出在测试期间保持不变。若不能实现此要求则省略该测试。

测试步骤：

1. TRANSMIT ReadProperty-Request,

‘Object Identifier’ = (the Loop object being tested),

‘Property Identifier’ = Manipulated\_Variable\_Reference

2. RECEIVE BACnet-ComplexACK-PDU,

‘Object Identifier’ = (the Loop object being tested),

‘Property Identifier’ = Manipulated\_Variable\_Reference,

‘Property Value’ = (any valid object property reference)

3. TRANSMIT ReadProperty-Request,

‘Object Identifier’ = (the Loop object being tested),

‘Property Identifier’ = Priority\_For\_Writing

4. RECEIVE BACnet-ComplexACK-PDU,

‘Object Identifier’ = (the Loop object being tested),

‘Property Identifier’ = Priority\_For\_Writing,

‘Property Value’ = (any priority from 1 to 16)

5. TRANSMIT ReadProperty-Request,

‘Object Identifier’ = (the Loop object being tested),

‘Property Identifier’ = Present\_Value

6. RECEIVE BACnet-ComplexACK-PDU,

‘Object Identifier’ = (the Loop object being tested),

‘Property Identifier’ = Present\_Value,

‘Property Value’ = (any valid value)

7. IF (the manipulated variable reference is commandable) THEN

VERIFY (the manipulated variable reference object),

(the referenced property) = (the Present\_Value from step 6),

ARRAY INDEX = (the Priority\_For\_Writing from step 4)

ELSE

VERIFY (the manipulated variable reference object),

(the referenced property) = (the Present\_Value from step 6)

1. Controlled\_Variable\_Reference跟踪

所需测试：ReadProperty服务执行测试，10.18。

BACnet参考条款：12.17.13。

目的：验证Controlled\_Variable\_Value是否跟踪Controlled\_Variable\_Reference属性改变。

配置要求：配置IUT，使测试循环对象的控制输出在测试期间保持不变。若不能实现则省略该测试。

测试步骤：

1. TRANSMIT ReadProperty-Request,

‘Object Identifier’ = (the Loop object being tested),

‘Property Identifier’ = Controlled\_Variable\_Reference

2. RECEIVE BACnet-ComplexACK-PDU,

‘Object Identifier’ = (the Loop object being tested),

‘Property Identifier’ = Controlled\_Variable\_Reference,

‘Property Value’ = (any valid object property reference)

3. TRANSMIT ReadProperty-Request,

‘Object Identifier’ = (the Loop object being tested),

‘Property Identifier’ = Controlled\_Variable\_Value

4. RECEIVE BACnet-ComplexACK-PDU,

‘Object Identifier’ = (the Loop object being tested),

‘Property Identifier’ = Controlled\_Variable\_Value,

‘Property Value’ = (any valid value)

5. VERIFY (the controlled variable reference object),

(the referenced property) = (the Controlled\_Variable\_Value from step 4)

1. Setpoint\_Reference跟踪

所需测试：ReadProperty服务执行测试，10.18。

BACnet参考条款：12.17.16。

目的：验证Setpoint是否跟踪Setpoint\_Reference属性改变。

配置要求：Loop对象Setpoint的值应根据Setpoint\_Reference属性中指定的对象和属性确定。除非特别指出，否则该参考设定点在测试期间应保持不变。如果不能实现这样的设置，则省略该测试。

测试步骤：

1. TRANSMIT ReadProperty-Request,

‘Object Identifier’ = (the Loop object being tested),

‘Property Identifier’ = Setpoint\_Reference

2. RECEIVE BACnet-ComplexACK-PDU,

‘Object Identifier’ = (the Loop object being tested),

‘Property Identifier’ = Setpoint\_Reference,

‘Property Value’ = (any valid object property reference)

3. TRANSMIT ReadProperty-Request,

‘Object Identifier’ = (the Loop object being tested),

‘Property Identifier’ = Setpoint

4. RECEIVE BACnet-ComplexACK-PDU,

‘Object Identifier’ = (the Loop object being tested),

‘Property Identifier’ = Setpoint,

‘Property Value’ = (any valid value)

5. VERIFY (the setpoint reference object),

(the referenced property) = (the setpoint from step 4)

6. IF (the referenced property of the setpoint reference object is writable) THEN

WRITE (the referenced property) = ( a different value)

ELSE

MAKE (the referenced property take on a new value)

7. VERIFY (the Loop object being tested),

Setpoint = (the new value of the referenced property)

1. 内部报告测试

验证循环对象的内部报告功能的测试将在9.4.5中介绍。

11. 多态输入对象测试
12. Out\_Of\_Service、Status\_Flags和Reliability测试

验证Out\_Of\_Service属性的功能以及Out\_Of\_Service、Status\_Flags和Reliability之间关系的测试已8.3.1.1中介绍。

1. Number\_Of\_States和State\_Text测试

所需测试：ReadProperty服务执行测试，10.18。

BACnet参考条款：12.18.11和12.18.12。

目的：验证State\_Text数组的大小是否与Number\_Of\_States相对应。

测试步骤：

1. TRANSMIT ReadProperty-Request,

‘Object Identifier’ = (the Multi-state Input object being tested),

‘Property Identifier’ = Number\_Of\_States

2. RECEIVE ReadProperty-ACK,

‘Object Identifier’ = (the Multi-state Input object being tested),

‘Property Identifier’ = Number\_Of\_States,

‘Property Value’ = (any integer greater than 0)

3. VERIFY State\_Text = (the number of states from step 2), ARRAY INDEX = 0

1. 内部报告测试

验证多态输入对象的内部报告功能的测试将在9.4.4中介绍。

1. 输入跟踪测试

所需测试：ReadProperty服务执行测试，10.18。

BACnet参考条款：12.18.4。

目的：验证跟踪和表示多状态输入值的能力。

配置要求：IUT应配置可在测试期间进行外部控制的多态输入，如果IUT无法配置，则省略该测试。

测试步骤：

1. REPEAT X = (at least two values meeting the functional range requirements of 7.2.1) DO {

MAKE (the real multi-state input X)

VERIFY Present\_Value = X

}

1. Number\_Of\_States和State\_Text大小变更测试

所需测试：WriteProperty服务执行测试，10.22

BACnet参考条款：12.18.11和12.18.12

目的：验证当Number\_Of\_States属性的值更改时，State\_Text数组的大小相应地更改为相同的大小。 如果无法更改Number\_Of\_States和State\_Text数组的大小，则不执行此测试。如果Protocol\_Revision不存在或者其值小于4，则不执行此测试。

配置要求：IUT应配置一个具有可写的Protocol\_Revision和可修改大小的State\_Text数组的多状态输入对象。

测试方案：Protocol\_Revision和State\_Text数组设置为特定值。然后通过写入状态数来增大，通过写入State\_Text数组来减小，通过写入State\_Text数组来增大，并通过写入状态数来减小。

1. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Multi-state Input object being tested),

‘Property Identifier’ = Number\_Of\_States,

‘Property Value’ = 2

2. RECEIVE Simple-ACK-PDU

3. VERIFY Number\_Of\_States = 2

4. VERIFY State\_Text = 2, ARRAY INDEX = 0

5. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Multi-state Input object being tested),

‘Property Identifier’ = Number\_Of\_States,

‘Property Value’ = (some value greater than 2)

6. RECEIVE Simple-ACK-PDU

7. VERIFY Number\_Of\_States = (the value written in step 5)

8. VERIFY State\_Text = (the value written in step 5), ARRAY INDEX = 0

9. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Multi-state Input object being tested),

‘Property Identifier’ = State\_Text,

‘Property Value’ = (State\_Text array of length 2)

10. RECEIVE Simple-ACK-PDU

11. VERIFY Number\_Of\_States = 2

12. VERIFY State\_Text = 2, ARRAY INDEX = 0

13. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Multi-state Input object being tested),

‘Property Identifier’ = State\_Text,

‘Property Value’ = (State\_Text array of length greater than 2)

14. RECEIVE Simple-ACK-PDU

15. VERIFY Number\_Of\_States = (the length of the array written in step 13)

16. VERIFY State\_Text = (the length of the array written in step 13), ARRAY INDEX = 0

17. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Multi-state Input object being tested),

‘Property Identifier’ = Number\_Of\_States,

‘Property Value’ = 2

18. RECEIVE Simple-ACK-PDU

19. VERIFY State\_Text = (an array consisting of elements 1 & 2 from the array written in step 13)

20. VERIFY Number\_Of\_States = 2

11. 多态输入对象测试
12. Out\_Of\_Service、Status\_Flags和Reliability测试

验证Out\_Of\_Service属性的功能以及Out\_Of\_Service、Status\_Flags和Reliability之间关系的测试已8.3.1.1中介绍。

1. Number\_Of\_States和State\_Text测试

所需测试：ReadProperty服务执行测试，10.18。

BACnet参考条款：12.19.11和12.19.12。

目的：验证State\_Text数组的大小是否与Number\_Of\_States相对应。

测试步骤：

1. TRANSMIT ReadProperty-Request,

‘Object Identifier’ = (the Multi-state Output object being tested),

‘Property Identifier’ = Number\_Of\_States

2. RECEIVE ReadProperty-ACK,

‘Object Identifier’ = (the Multi-state Output object being tested),

‘Property Identifier’ = Number\_Of\_States,

‘Property Value’ = (any integer greater than 0)

3. VERIFY State\_Text = (the number of states from step 2), ARRAY INDEX = 0

1. 优先级命令测试

验证多态输出对象的优先级命的测试已在8.3.1.2和 8.3.1.3中介绍。

1. 内部报告测试

验证多态输出对象的内部报告功能的测试将在9.4.4中介绍。

1. 输出跟踪测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.19.4。

目的：验证跟踪和表示多状态输出值的能力。

配置要求：IUT应配置可在测试期间观测到的多态输出，如果IUT无法配置，则省略该测试。

测试步骤：

1. REPEAT X = (all values meeting the functional range requirements of 7.2.1) DO {

WRITE Present\_Value = X

VERIFY Present\_Value = X

CHECK (verify that the output is X)

}

1. Number\_Of\_States和State\_Text大小变更测试

所需测试：WriteProperty服务执行测试，10.22

BACnet参考条款：12.19.11和12.19.12

测试步骤：

验证多状态输出对象的Number\_Of\_States和State\_Text数组大小的测试已在8.3.2.18.5中定义。使用多状态输出对象进行测试。

1. 多态值对象测试
2. Out\_Of\_Service、Status\_Flags和Reliability测试

验证Out\_Of\_Service属性的功能以及Out\_Of\_Service、Status\_Flags和Reliability之间关系的测试已在8.3.1.1中介绍。

1. Number\_Of\_States和State\_Text测试

所需测试：ReadProperty服务执行测试，10.18。

BACnet参考条款：12.20.11和12.20.12。

目的：验证State\_Text数组的大小是否与Number\_Of\_States相对应。

测试步骤：

1. TRANSMIT ReadProperty-Request,

‘Object Identifier’ = (the Multi-state Value object being tested),

‘Property Identifier’ = Number\_Of\_States

2. RECEIVE ReadProperty-ACK,

‘Object Identifier’ = (the Multi-state Value object being tested),

‘Property Identifier’ = Number\_Of\_States,

‘Property Value’ = (any integer greater than 0)

3. VERIFY State\_Text = (the number of states from step 2), ARRAY INDEX = 0

1. 优先级命令测试

验证多态输出对象的优先级命令的测试已在8.3.1.2和8.3.1.3中介绍。

1. 内部报告测试

验证多态输出对象的内部报告功能的测试将在9.4.2中介绍。

1. Number\_Of\_States和State\_Text大小变更测试

所需测试：WriteProperty服务执行测试，10.22

BACnet参考条款：12.20.10和12.20.11

测试步骤：

验证多态值对象的Number\_Of\_States和State\_Text数组大小的测试已在8.3.2.18.5中定义。使用多状态输出对象进行测试。

1. 通知类对象
2. 优先级测试

所需测试：ConfirmedEventNotification服务启动测试，9.4; UnconfirmedEventNotification服务启动测试，9.5；ReadProperty服务执行测试，10.18; WriteProperty服务执行测试，10.22。

BACnet参考条款：12.21.6。

目的：验证在启动偶数通知时IUT是否实现通知类对象的优先级属性的功能。

测试方案：TD将选择通知类对象的一个实例和链接到它的事件生成对象的一个实例。设置事件生成对象属性，使Event\_State分别从NORMAL更改为OFFNORMAL，从OFFNORMAL更改为NORMAL，从NORMAL更改为FAULT，以及从FAULT更改为NORMAL。验证每个状态转换的结果事件通知的优先级的正确性。事件必须可被触发，否则测试失败。

配置要求：IUT应配置一个或多个通知类对象实例并且在至少有一个事件生成对象链接到通知类对象。事件生成对象可以是支持内部报告的任何对象，或者它可以是Event Enrollment对象。通知类对象应配置独立的、不同的优先级值（用于TO-OFFNORMAL，TO-NORMAL和TO-FAULT转换）。所有事件使能位都应设置为TRUE。引用的事件触发属性值应使对象处在NORMAL态。

在下面的测试描述中，“X”表示指定事件触发属性。

测试步骤：

1. WAIT (Time\_Delay + **Notification Fail Time**)

2. VERIFY Event\_State = NORMAL

3. IF (X is writable) THEN

WRITE X = (a value that is OFFNORMAL)

ELSE

MAKE (X have a value that is OFFNORMAL)

4. WAIT (Time\_Delay)

5. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the event-generating object configured for this test),

‘Time Stamp’ = (any valid time stamp),

‘Notification Class’ = (the class corresponding to the object being tested),

‘Priority’ = (the value configured to correspond to a TO- OFFNORMAL transition),

‘Event Type’ = (any valid event type),

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = OFFNORMAL,

‘Event Values’ = (values appropriate to the event type)

6. VERIFY Event\_State = OFFNORMAL

7. IF (X is writable) THEN

WRITE X = (a value that is NORMAL)

ELSE

MAKE (X have a value that is NORMAL)

8. WAIT (Time\_Delay)

9. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the event-generating object configured for this test),

‘Time Stamp’ = (any valid time stamp),

‘Notification Class’ = (the class corresponding to the object being tested),

‘Priority’ = (the value configured to correspond to a TO-NORMAL transition),

‘Event Type’ = (any valid event type),

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = OFNORMAL,

‘To State’ = NORMAL,

‘Event Values’ = (values appropriate to the event type)

10. VERIFY Event\_State = NORMAL

11. IF (the event-triggering object can be placed into a fault condition) THEN {

12. MAKE (the event-triggering object change to a fault condition)

13. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the event-generating object configured for this test),

‘Time Stamp’ = (any valid time stamp),

‘Notification Class’ = (the class corresponding to the object being tested),

‘Priority’ = (the value configured to correspond to a TO-FAULT transition),

‘Event Type’ = (any valid event type),

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = FAULT,

‘Event Values’ = (values appropriate to the event type)

14. VERIFY Event\_State = FAULT

15. MAKE (the event-triggering object change to a normal condition)

16. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the event-generating object configured for this test),

‘Time Stamp’ = (any valid time stamp),

‘Notification Class’ = (the class corresponding to the object being tested),

‘Priority’ = (the value configured to correspond to a TO- NORMAL transition),

‘Event Type’ = (any valid event type),

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = FAULT,

‘To State’ = NORMAL,

‘Event Values’ = (values appropriate to the event type)

17. VERIFY Event\_State = NORMAL

}

注意事项：可以用UnconfirmedEventNotification服务替换ConfirmedEventNotification服务。测试描述中省略了‘Message Text’参数，因为它是可选的，IUT报告消息中可能包含此该参数。

1. Ack\_Required测试

这些测试验证通知类“Ack\_Required”属性的值是否正确反映在事件通知消息的发布中。

1. Ack\_Required False测试

所需测试：ConfirmedEventNotification服务启动测试，9.4; UnconfirmedEventNotification服务启动测试，9.5;ReadProperty服务执行测试，10.18; WriteProperty服务执行测试，10.22。

BACnet参考条款：12.21.7。

目的：验证过如果Ack\_Required属性指示事件通知不需要确认，则通知消息的AckRequired参数可传递该指示。如果IUT不支持未确认的报告通知，则省略该测试。

测试方案：TD将选择通知类对象的一个实例和链接到它的事件生成对象的一个实例。设置事件生成对象属性，使事件状态分别从NORMAL更改为OFFNORMAL，从OFFNORMAL更改回NORMAL，从NORMAL更改为FAULT，以及从FAULT更改回NORMAL。验证每个状态转换的“AckRequired”参数值。

配置要求：配置要求与8.3.2.20.1相同，除了Ack\_Required属性的值应为B’000’，表示不需确认。

测试步骤：测试步骤与8.3.2.20.1相同，除了“AckRequired”参数在所有事件通知消息中都应为FALSE。

注意事项：UnconfirmedEventNotification服务可以替换ConfirmedEventNotification服务。测试描述中省略了‘Message Text’参数，因为它是可选的，IUT通知消息中可能包括此参数。

1. Ack\_Required True测试

所需测试：ConfirmedEventNotification服务启动测试，9.4; Unc onfirmedEventNotification服务启动测试，9.5;ReadProperty服务执行测试，10.18; WriteProperty服务执行测试，10.22。

BACnet参考条款：12.21.7。

目的：验证过如果Ack\_Required属性指示事件通知需要确认，则通知消息的AckRequired参数可传递该指示。如果IUT不支持确认的报告通知，则省略该测试。

测试方案：TD将选择通知类对象的一个实例和链接到它的事件生成对象的一个实例。设置事件生成对象属性，使事件状态分别从NORMAL更改为OFFNORMAL，从OFFNORMAL更改回NORMAL，从NORMAL更改为FAULT，以及从FAULT更改回NORMAL。验证每个状态转换的“AckRequired”参数值。

配置要求：配置要求与8.3.2.20.1相同，除了Ack\_Required属性的值应为B’111’，表示需要确认。

测试步骤：测试步骤与8.3.2.20.1相同，除了“AckRequired”参数在所有事件通知消息中都应为TRUE。

注意事项：UnconfirmedEventNotification服务可以替换ConfirmedEventNotification服务。测试描述中省略了‘Message Text’参数，因为它是可选的，IUT通知消息中可能包括此参数。

1. Recipient\_List测试

验证Recipient\_List属性中BACnetDestination项的各参数是否正确。

1. ValidDays测试

所需测试：ConfirmedEventNotification服务启动测试，9.4; UnconfirmedEventNotification服务启动测试，9.5; ReadProperty服务执行测试，10.18; WriteProperty服务执行测试，10.22; 时间同步服务执行测试，10.30。

BACnet参考条款：12.21.8。

目的：验证在通知类对象的Recipient\_List属性中使用的BACnetDestination的Valid Days参数的操作。

测试方案：TD选择通知类对象的一个实例和链接到它的事件生成对象的一个实例。通知类对象的Recipient\_List应包含一个配置了Valid Days参数的收件人，至少一天为TRUE，且至少一天为FALSE。将操作设置事件生成对象的属性以使Event\_Enable从NORMAL更改为OFFNORMAL。验证如果本地日期是Valid Days之一，则发送通知消息，如果本地日期不是有效日期，则不发送通知消息。

配置要求：IUT配置一个或多个通知类对象实例和至少一个链接到通知类对象的事件生成对象。事件生成对象可以是任何支持内部报告的对象，或Event Enrollment对象。设置事件生成对象的Event\_Enable属性，使其可传输所有事件转换的通知消息。设置事件生成对象事件初始态为NORMAL。通知类对象应配置Recipient\_List中的一个收件人。设置Valid Days参数，使一周中至少有一天的值为TRUE，至少有一天值为FALSE。设置接收方的Transitions参数，以接收所有事件转换的通知。

在下述测试，“X”表示指定事件触发属性。

测试步骤：

1. (TRANSMIT TimeSynchronization-Request,

‘Time’ = (any time within the window defined by From Time and To Time in the BACnet Destination that corresponds to one of the valid days)) |

MAKE (the local date and time = (any time within the window defined by From Time and To Time in the BACnetDestination that corresponds to one of the valid days))

2. WAIT (Time\_Delay + **Notification Fail Time**)

3. VERIFY Event\_State = NORMAL

4. IF (X is writable) THEN

WRITE X = (a value that is OFFNORMAL)

ELSE

MAKE (X have a value that is OFFNORMAL)

5. WAIT (Time\_Delay)

6. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the event-generating object configured for this test),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the class corresponding to the object being tested),

‘Priority’ = (the value configured to correspond to a TO- OFFNORMAL transition),

‘Event Type’ = (any valid event type),

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = OFFNORMAL,

‘Event Values’ = (values appropriate to the event type)

7. VERIFY Event\_State = OFFNORMAL

8. (TRANSMIT TimeSynchronization-Request,

‘Time’ = (any time within the window defined by From Time and To time in the BACnet Destination that corresponds to one of the invalid days)) |

MAKE (the local date and time = (any time within the window defined by From Time and To Time in the BACnetDestination that corresponds to one of the invalid days))

9. IF (X is writable) THEN

WRITE X = (a value that is NORMAL)

ELSE

MAKE (X have a value that is NORMAL)

10. WAIT (Time\_Delay + **Notification Fail Time**)

11. CHECK (verify that no notification message was transmitted)

注意事项：UnconfirmedEventNotification服务可以替换ConfirmedEventNotification服务。测试描述中省略了Message Text参数，因为它是可选的，IUT通知消息中可能包括此参数。

1. FromTime和ToTime测试

所需测试：Valid Days测试，8.3.2.21.3.1；ConfirmedEventNotification服务启动测试，9.4;UnconfirmedEventNotification服务启动测试，9.5；ReadProperty服务执行测试，10.18; TimeSynchronization服务执行测试，10.30。

BACnet参考条款：12.21.8。

目的：验证通知类对象的Recipient\_List属性中使用的BACnetDestination的From Time和To Time参数的操作。

测试方案：对于本地日期和时间范围由From Time和To Time参数定义的情况，8.3.2.21.3.1中的ValidDays测试已包括在内。此测试使用相同的IUT配置，并将本地时间设置为一个ValidDays值，但不在由From time和to time参数定义的窗口内的目的是验证在触发事件时不传输事件通知消息。

配置要求：与8.3.2.21.3.1相同。

测试步骤：

1. (TRANSMIT TimeSynchronization-Request,

‘Time’ = (any time outside the window defined by From Time and To Time in the BACnet Destination that corresponds to one of the valid days)) |

MAKE (the local date and time = (any time outside the window defined by From Time and To Time in the BACnetDestination that corresponds to one of the valid days))

2. WAIT (Time\_Delay + **Notification Fail Time**)

3. VERIFY Event\_State = NORMAL

4. IF (X is writable) THEN

WRITE X = (a value that is OFFNORMAL)

ELSE

MAKE (X have a value that is OFFNORMAL)

5. WAIT (Time\_Delay + **Notification Fail Time**)

6. CHECK (verify that no notification message was transmitted)

1. IssueConfirmedNotifications测试

所需测试：ConfirmedEventNotification服务启动测试，9.4；UnconfirmedEventNotification服务启动测试，9.5； WriteProperty服务执行测试，10.22。

BACnet参考条款：12.21.8。

目的：如果IssueConfirmedNotifications参数的值为TRUE，验证是否使用了ConfirmedEventNotification消息；如果值为FALSE，则使用UnconfirmedEventNotification消息。 如果IUT不支持确认和未确认事件通知，则省略此测试。

配置要求：IUT应配置两个或多个通知类对象实例和链接到通知类对象的事件生成对象。事件生成对象可以是支持内部报告的对象，或Event Enrollment对象。设置事件生成对象的Event\_Enable属性，使其可对所有事件转换发送通知消息。事件生成对象的事件初始态为NORMAL。通知类对象N1的发布确认通知为TRUE，通知类对象N2发布确认通知为FALSE。设置Valid Days使一周中至少有一天的值为TRUE。设置接收者的Transitions参数，以接收所有事件转换的通知。本地日期和时间范围为由From Time和To Time定义的ValidDays的一个。

在下述测试中，X1、X2分别表示通知对象N1、N2的事件触发属性。

测试步骤：

1. VERIFY (the event-generating object linked to N1), Event\_State = NORMAL

2. VERIFY (the event-generating object linked to N2), Event\_State = NORMAL

3. WAIT (Time\_Delay + **Notification Fail Time**)

4. IF (X1 is writable) THEN

WRITE X1 = (a value that is OFFNORMAL)

ELSE

MAKE (X1 a value that is OFFNORMAL)

5. WAIT (Time\_Delay)

6. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the event-generating object linked to N1),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the class corresponding to the object being tested),

‘Priority’ = (the value configured to correspond to a TO- OFFNORMAL transition),

‘Event Type’ = (any valid event type),

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = OFFNORMAL,

‘Event Values’ = (values appropriate to the event type)

7. IF (X2 is writable) THEN

WRITE X2 = (a value that is OFFNORMAL)

ELSE

MAKE (X2 a value that is OFFNORMAL)

8. WAIT (Time\_Delay)

9. BEFORE **Notification Fail Time**

RECEIVE UnconfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the event-generating object linked to N2),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the class corresponding to the object being tested),

‘Priority’ = (the value configured to correspond to a TO- OFFNORMAL transition),

‘Event Type’ = (any valid event type),

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = OFFNORMAL,

‘Event Values’ = (values appropriate to the event type)

注意事项：如果Recipient\_List可写且可以更改IssueConfirmedNotifications，则通过写入Recipient\_List，可以仅使用一个通知类对象执行此测试，以便在确认和未确认通知之间进行改变。 测试描述中省略了Message Text参数，因为它是可选的，IUT通知消息中可能包括该参数。

1. 转换测试

所需测试：ConfirmedEventNotification服务启动测试，9.4; UnconfirmedEventNotification服务启动测试，9.5;ReadProperty服务执行测试，10.18; WriteProperty服务执行测试，10.22。

BACnet参考条款：12.21.8。

目的：验证只有Transitions参数中的位与事件转换对应时才能传送通知消息。

测试方案：配置Transitions参数，说明哪些事件转换触发事件通知而哪些不触发。触发每个事件转换，监测IUT，验证只有Transitions参数值为TRUE的事件转换发生时才发送通知消息。

配置要求：IUT配置一个或多个通知类对象实例和至少一个链接到通知类对象的事件生成对象。事件生成对象可以是任何支持内部报告的对象，或Event Enrollment对象。设置事件生成对象的Event\_Enable属性，使其可对所有事件转换发送通知消息。事件生成对象的事件初始态为NORMAL。通知类对象配置Recipient\_List中的一个收件人。Transitions参数值为TRUE。TO-OFFNORMAL转换、TO-ORMAL和其他事件转换值为FALSE。本地时间范围为由From Time和To Time定义的有效日期。

在下面的测试描述中，“X”用于指定事件触发属性。

1. VERIFY Event\_State = NORMAL

2. WAIT (Time\_Delay + **Notification Fail Time**)

3. IF (X is writable) THEN

WRITE X = (a value that is OFFNORMAL)

ELSE

MAKE (X have a value that is OFFNORMAL)

4. WAIT (Time\_Delay)

5. BEFORE **Notification Fail Time**

IF (the Transitions bit corresponding to the TO-OFFNORMAL transition is TRUE) THEN

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the event-generating object configured for this test),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the class corresponding to the object being tested),

‘Priority’ = (the value configured to correspond to a TO- OFFNORMAL transition),

‘Event Type’ = (any valid event type),

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = OFFNORMAL,

‘Event Values’ = (values appropriate to the event type)

ELSE

CHECK (verify that the IUT did not transmit an event notification message)

6. VERIFY Event\_State = OFFNORMAL

7. IF (X is writable) THEN

WRITE X = (a value that is NORMAL)

ELSE

MAKE (X have a value that is NORMAL)

8. WAIT (Time\_Delay)

9. BEFORE **Notification Fail Time**

IF (the Transitions bit corresponding to the TO-NORMAL transition is TRUE) THEN

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the event-generating object configured for this test),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the class corresponding to the object being tested),

‘Priority’ = (the value configured to correspond to a TO-NORMAL transition),

‘Event Type’ = (any valid event type),

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = OFFNORMAL,

‘To State’ = NORMAL,

‘Event Values’ = (values appropriate to the event type)

ELSE

CHECK (verify that the IUT did not transmit an event notification message)

10. VERIFY Event\_State = NORMAL

11. IF (the event-triggering object can be placed into a fault condition) THEN {

MAKE (the event-triggering object change to a fault condition)

BEFORE **Notification Fail Time**

IF (the Transitions bit corresponding to the TO-FAULT transition is TRUE) THEN

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the event-generating object configured for this test),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the class corresponding to the object being tested),

‘Priority’ = (the value configured to correspond to a TO- FAULT transition),

‘Event Type’ = (any valid event type),

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = FAULT,

‘Event Values’ = (values appropriate to the event type)

ELSE

CHECK (verify that the IUT did not transmit an event notification message)

VERIFY Event\_State = FAULT

}

注意事项：UnconfirmedEventNotification服务可以替换ConfirmedEventNotification服务。测试描述中省略了Message Text参数，因为它是可选的，IUT通知消息中可能包括此参数。

1. Recipient\_List 属性支持设备标识符收件人测试

目的：验证通知类对象的Recipient\_List属性是否支持接受方组件的设备形式，及IUT是否能够将MAC地址与设备标识符相关联。以确保IUT能够找到指定的警报接收者并向指定的接收者发送通知。如果IUT的通知类对象的Recipient\_List属性支持BACnet收件人的BACnet对象标识符形式，则进行此测试。

测试方案：在IUT中选择一个引用通知类对象N的事件生成对象E，测试人员应在相关通知类对象的Recipient\_List中添加一个条目，该对象为IUT不知道的设备指定设备标识符D。作为设备D的TD应位于与IUT不同的网络上，以确保IUT能够连接到位于任何网络上的接收者。

配置要求：TD不执行WhoHas。

测试步骤：

1. WRITE N.RecipientList = ( {all days, all times, D, any process ID, FALSE, all transitions} )

2. MAKE (the event generating object, E, transition)

3. BEFORE **Notification Fail Time** plus the amount of time the IUT takes to perform device discovery

RECEIVE UnconfirmedEventNotification-Request,

‘Process Identifier’ = (the valid process ID from step 1),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = E,

‘Time Stamp’ = (any valid time stamp),

‘Notification Class’ = (N’s instance),

‘Priority’ = (any valid priority),

‘Event Type’ = (any valid event type),

‘Notify Type’ = ALARM | EVENT,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = (any valid event state),

‘To State’ = (any valid event state),

‘Event Values’ = (values appropriate to the event type)

注意事项：修改Recipient\_List、发送通知之前，IUT发起一个或多个范围限制的WhoIs请求。IUT可能还需要执行其他操作寻找网络。由于有多种方法可以使用WhoIs寻找设备，测试只关注IUT查找设备D的能力，而不关注WhoIs请求的具体内容和时间。

1. Recipient\_List 属性支持网络地址收件人测试

目的：验证通知类对象的Recipient\_List属性是否支持接收组件的地址形式。目的是确保IUT能够向指定的收件人发送通知。

测试方案：在IUT中选择一个引用通知类对象N的事件生成对象E，测试人员应在相关通知类对象的Recipient\_List中添加一个条目，该对象指定BACnet地址A，其中A是单播或本地、远程或全局广播地址。

测试步骤：

1. WRITE N.RecipientList = ( {all days, all times, A, any process ID, FALSE, all transitions} )

2. MAKE (the event generating object, E, transition)

3. BEFORE **Notification Fail Time**

RECEIVE

DESTINATION = A,

UnconfirmedEventNotification-Request,

‘Process Identifier’ = (the valid process ID from step 1),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = E,

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (N’s instance),

‘Priority’ = (any valid priority),

‘Event Type’ = (any valid event type),

‘Notify Type’ = ALARM | EVENT,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = (any valid event state),

‘To State’ = (any valid event state),

‘Event Values’ = (values appropriate to the event type)

1. Program程序对象测试

程序对象的自定义程序网络的参数可见。由于BACnet没有定义程序的功能，因此没有标准测试来验证此功能。

1. Schedule时间表对象测试

以下测试针对属性不要求可写或可配置的时间表对象。但如果时间表对象可以任意方式配置，则其配置应符合以下测试。如果不能以测试所需的方式去配置IUT，则省略测试。如果IUT支持可以在设备外部写入的时间表对象，则应在其中一个时间表测试中进行测试。

时间表对象的测试预定在特定日期和时间进行的写操作，并通过读取时间表对象的Present\_Value属性来验证。为在合适的时间内执行测试，设备的时钟和日历需可改。

使用预定写入操作的测试，应预先确定写入的日期和时间。表8-1至8-9给出了在测试中使用的日期（记为D1，D2等）的标准。满足这些标准的日期可以从现有的时间表中选择，或者由供应商制定时间表。

Vn与BACnetTimeValue对中的日期Dn中的时间相关，可取任意原始数据类型。

1. Effective\_Period测试

所需测试：ReadProperty服务执行测试，10.18; TimeSynchronization服务执行测试，10.30。

BACnet参考条款：12.24.6。

目的：验证时间表有效时，有效期是否控制日期范围。

测试方案：TD根据表8-1中的标准选择两个日期值，其中一个在有效期之外，另一个则处于有效期内的已知预定状态。IUT的本地日期和时间在这两个日期中变化，监测Present\_Value属性，验证写操作仅在有效期内进行。

配置要求：IUT配置一个时间表对象，使表8-1中定义的时间段具有唯一的时间表值。设置本地日期和时间，使Present\_Value属性值为除了V1外的值。

**表8-1** Effective\_Period测试日期和值的标准

|  |  |  |
| --- | --- | --- |
| **日期：** | **标准：** | **值：** |
| D1 | 1. Date在Effective\_Period内发生  2. Date在Weekly\_Schedule和Exception\_Schedule有效 | V1 |
| D2 | 1. Date不在Effective\_Period内进行  2. Date在Weekly\_Schedule和Exception\_Schedule有效 | 不同于V1的V2 |

测试步骤：

1. VERIFY Present\_Value = (any value other than V1)

2. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D1) | MAKE (the local date and time = D1)

3. WAIT **Schedule** **Evaluation Fail Time**

4. VERIFY Present\_Value = V1

5. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D2) | MAKE (the local date and time = D2)

6. WAIT **Schedule** **Evaluation Fail Time**

7. VERIFY Present\_Value = V2

1. Weekly\_Schedule属性测试

所需测试：ReadProperty服务执行测试，10.18; TimeSynchronization服务执行测试，10.30。

BACnet参考条款：12.24.7。

目的：验证Weekly\_Schedule是否包括一周中每一天的可区分计划，以及一天的整个计划是否可执行。

测试方案：IUT的本地日期和时间按表8-2所示顺序更改，设置本地日期和时间。对Present\_Value属性进行监视，以验证每个单独计划的日期是否发生了写入操作。

配置要求：IUT应配置一个计划对象，该对象包含一个周计划，七个满足表8-2的要求的可区分的日计划。本地日期和时间的设置应确保Present\_Value属性的值不是V1。如果没有满足这些要求且无法进行配置，则应省略此测试。

**表8-2** Weekly\_Schedule测试日期和值的标准

|  |  |  |
| --- | --- | --- |
| **日期：** | **标准：** | **值：** |
| D1 | 1. Date在Effective\_Period内发生  2. Date周一发生  3. Date在Exception\_Schedule内无效 | V1 |
| D2 | 1. Date在Effective\_Period内发生  2. Date周二发生  3. Date在Exception\_Schedule内无效 | V2，不等于V1 |
| D3 | 1. Date在Effective\_Period内发生  2. Date周三发生  3. Date在Exception\_Schedule内无效 | V3，不等于V2 |
| D4 | 1. Date在Effective\_Period内发生  2. Date周四发生  3. Date在Exception\_Schedule内无效 | V4，不等于V3 |
| D5 | 1. Date在Effective\_Period内发生  2. Date周五发生  3. Date在Exception\_Schedule内无效 | V5，不等于V4 |
| D6 | 1. Date在Effective\_Period内发生  2. Date周六发生  3. Date在Exception\_Schedule内无效 | V6，不等于V5 |
| D7 | 1. Date在Effective\_Period内发生  2. Date周日发生  3. Date在Exception\_Schedule内无效 | V7，不等于V6 |

测试步骤：

1. VERIFY Present\_Value = (any value other than V1)

2. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D1) | MAKE (the local date and time = D1)

3. WAIT **Schedule Evaluation Fail Time**

4. VERIFY Present\_Value = V1

5. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D2) | MAKE (the local date and time = D2)

6. WAIT **Schedule Evaluation Fail Time**

7. VERIFY Present\_Value = V2

8. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D3) | MAKE (the local date and time = D3)

9. WAIT **Schedule Evaluation Fail Time**

10. VERIFY Present\_Value = V3

11. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D4) | MAKE (the local date and time = D4)

12. WAIT **Schedule Evaluation Fail Time**

13. VERIFY Present\_Value = V4

14. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D5) | MAKE (the local date and time = D5)

15. WAIT **Schedule Evaluation Fail Time**

16. VERIFY Present\_Value = V5

17. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D6) | MAKE (the local date and time = D6)

18. WAIT **Schedule Evaluation Fail Time**

19. VERIFY Present\_Value = V6

20. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D7) | MAKE (the local date and time = D7)

21. WAIT **Schedule Evaluation Fail Time**

22. VERIFY Present\_Value = V7

23. REPEAT X = (the time portion of the BACnetTimeValue entries for one of the daily schedules in Table 7-2) DO {

(TRANSMIT TimeSynchronization-Request, ‘Time’ = X) | MAKE (the local date and time = X)

WAIT **Schedule Evaluation Fail Time**

VERIFY Present\_Value = (the scheduled value corresponding to time X)

}

1. Exception\_Schedule属性测试

如果IUT不适配本节测试，则应省略此测试。无法进行配置可能是由于Exception\_Schedule属性不存在或不可变，Exception\_Schedule中BACnetSpecialEvents可用数量有限，或Calendar对象不可用。

1. 日历参考测试
2. *参见7.3.2.23.X.2.1节校订4日历参考测试*。日历录入日期测试

所需测试：ReadProperty服务执行测试，10.18;TimeSynchronization服务执行测试，10.30。

BACnet参考条款：12.24.8。

目的：验证Exception\_Schedule中的指定日期能否启用引用时间表对象。

测试方案：TD根据表8-3标准更改IUT本地日期和时间，监测Present\_Value属性，验证是否发生写操作。

配置要求：IUT应包含时间表对象和Exception\_Schedule，且Exception\_Schedule包含具有特定日期的BACnetCalendarEntry。表8-3给出了测试中使用日期的标准。设置本地日期和时间，使Present\_Value属性值不为V1。如果设备对象有Protocol\_Revision属性且值大于等于4，则不执行此测试。

**表8-3** 日历录入日期和值的标准

|  |  |  |
| --- | --- | --- |
| **日期：** | **标准：** | **值：** |
| D1 | 1. Date在Effective\_Period内进行  2A. BACnetSpecialEvent包含calendarEntry：Date  2B. Date与calendarEntry匹配  2C.比其他正在进行的BACnetSpecialEvent有更高的事件优先级 | V1 |
| D2 | 1. Date在Effective\_Period内进行  2. Date不出现在任何BACnetSpecialEvent中  3.与引用时间表对象的BACnetDailtySchedule同一日期进行，时间紧随其后 | 不等于V1 |

测试步骤：

1. VERIFY Present\_Value = (any value other than V1)

2. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D1) | MAKE (the local date and time = D1)

3. WAIT **Schedule Evaluation Fail Time**

4. VERIFY Present\_Value = V1

5. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D2) | MAKE (the local date and time = D2)

6. WAIT **Schedule Evaluation Fail Time**

7. VERIFY Present\_Value = (any value other than V1)

1. 日历录入DateRange测试

所需测试：ReadProperty服务执行测试，10.18;TimeSynchronization服务执行测试，10.30。

BACnet参考条款：12.24.8。

目的：验证Exception\_Schedule日期范围中的日期能否启用引用时间表对象。

测试方案：TD根据表8-4标准更改IUT本地日期和时间，监测Present\_Value属性，验证是否发生写操作。

配置要求：IUT应包含时间表对象和Exception\_Schedule，且Exception\_Schedule包含有日期范围的BACnetCalendarEntry。表8-4给出了测试中使用日期的标准。设置本地日期和时间，使Present\_Value属性值不为V1。如果设备对象有 Protocol\_Revision属性且值大于等于4，则不执行此测试。

**表8-4** 日历录入DateRange测试日期和值的标准

|  |  |  |
| --- | --- | --- |
| **日期：** | **标准：** | **值：** |
| D1 | 1. Date在Effective\_Period内进行  2A. BACnetSpecialEvent包含calendarEntry：DateRange  2B. Date与BACnetCalendarEntry: DateRange  2C.比其他正在进行的BACnetSpecialEvents有更高的事件优先级 | V1 |
| D2 | 1. Date在Effective\_Period内进行  2.不出现在任何BACnetSpecialEvent中  3.与引用时间表对象的BACnetDailtySchedule同一日期进行，时间紧随其后 | 不等于V1 |

测试步骤：

1. VERIFY Present\_Value = (any value other than V1)

2. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D1) | MAKE (the local date and time = D1)

3. WAIT **Schedule Evaluation Fail Time**

4. VERIFY Present\_Value = V1

5. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D2) | MAKE (the local date and time = D2)

6. WAIT **Schedule Evaluation Fail Time**

7. VERIFY Present\_Value = (any value other than V1)

1. 日历录入WeekNDay Month测试

所需测试：ReadProperty服务执行测试，10.18;TimeSynchronization服务执行测试，10.30。

BACnet参考条款：12.24.8。

目的：验证Exception\_Schedule中与WeekNDayMonth字段匹配的日期能否启用引用时间表对象。

测试方案：TD根据表8-5中标准更改IUT本地日期和时间，检测Present\_Value属性，验证是否发生了计划的写操作。

配置要求：IUT包含一个时间表对象，其Exception\_Schedule包含一个BACne tCalendarEntry，其中WeekNDay录入指定为月。表8-5给出了测试中使用日期的标准。设置本地日期和时间，使Present\_Value属性值不为V1。如果设备对象有Protocol\_Revision属性且值大于等于4，则不执行此测试。

**表8-5** 日历录入WeekNDay Month测试日期和值的标准

|  |  |  |
| --- | --- | --- |
| **日期：** | **标准：** | **值：** |
| D1 | 1. Date在Effective\_Period内进行  2A. BACnetSpecialEvent包含calendarEntry：WeekNDay  2B: calendarEntry：WeekNDay：指定月份  2C. Date与calendarEntry匹配：WeekNDay： Month,  2D.比其他正在进行的BACnet SpecialEvents有更高的事件优先级 | V1 |
| D2 | 1. Date在Effective\_Period内进行  2.不出现在任何BACnetSpecialEvents中  3. 与引用时间表对象的BACnetDailtySchedule同一日期进行，时间紧随其后 | 不等于V1 |

测试步骤：

1. VERIFY Present\_Value = (any value other than V1)

2. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D1) | MAKE (the local date and time = D1)

3. WAIT **Schedule Evaluation Fail Time**

4. VERIFY Present\_Value = V1

5. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D2) | MAKE (the local date and time = D2)

6. WAIT **Schedule Evaluation Fail Time**

7. VERIFY Present\_Value = (any value other than V1)

1. 日历录入WeekNDay WeekOfMonth测试

所需测试：ReadProperty服务执行测试，10.18;TimeSynchronization服务执行测试，10.30。

BACnet参考条款：12.24.8。

目的：验证Exception\_Schedule中与WeekNDay的WeekOfMonth匹配的日期能否启用引用时间表对象。

测试方案：TD根据表8-6中标准更改IUT本地日期和时间，检测Present\_Value属性，验证是否发生了计划的写操作。

配置要求：IUT包含一个时间表对象，其Exception\_Schedule包含一个BACnetCalendarEntry，其中WeekNDay录入指定为某月的某周。表8-6给出了测试中使用日期的标准。设置本地日期和时间，使Present\_Value属性值不为V1。如果设备对象有 Protocol\_Revision属性且值大于等于4，则不执行此测试。

**表8-6** 日历录入WeekNDay的WeekOfMonth测试日期和值的标准

|  |  |  |
| --- | --- | --- |
| **日期：** | **标准：** | **值：** |
| D1 | 1.在有效期内进行  2A. BACnet特殊事件包含 calendarEntry：WeekNDay  2B: calendarEntry：WeekNDay：指定WeekOfMonth  2C: calendarEntry：WeekNDay：指定WeekOfMonth范围  2D.日期与 calendarEntry匹配：WeekNDay：WeekOfMonth  2E.比其他正在进行的BACnet SpecialEvents有更高的事件优先级 | V1 |
| D2 | 1. 在有效期内进行  2A. BACnet SpecialEvents包含 calendarEntry：WeekNDay  2B: calendarEntry：WeekNDay：指定WeekOfMonth  2C: calendarEntry：WeekNDay：指定WeekOfMonth范围1-5  2D. 日期不与 calendarEntry匹配：WeekNDay：WeekOfMonth | 不等于V1 |

测试步骤：

1. VERIFY Present\_Value = (any value other than V1)

2. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D1) | MAKE (the local date and time = D1)

3. WAIT **Schedule Evaluation Fail Time**

4. VERIFY Present\_Value = V1

5. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D2) | MAKE (the local date and time = D2)

6. WAIT **Schedule Evaluation Fail Time**

7. VERIFY Present\_Value = (any value other than V1)

1. 日历录入WeekNDay Last WeekOfMonth测试

所需测试：ReadProperty服务执行测试，10.18;TimeSynchronization服务执行测试，10.30。

BACnet参考条款：12.24.8。

目的：验证Exception\_Schedule中与WeekNDay的WeekOfMonth匹配的日期能否启用引用时间表对象。

测试方案：TD根据表8-7中标准更改IUT本地日期和时间，检测Present\_Value属性，验证是否发生了计划的写操作。

配置要求：IUT包含一个时间表对象，其Exception\_Schedule包含一个BACnetCalendarEntry，其中WeekNDay录入指定为某月的最后一周。表8-7给出了测试中使用日期的标准。设置本地日期和时间，使Present\_Value属性值不为V1。如果设备对象有 Protocol\_Revision属性且值大于等于4，则不执行此测试。

**表8-7** 日历录入WeekNDay的Last WeekOfMonth测试日期和值的标准

|  |  |  |
| --- | --- | --- |
| **日期：** | **标准：** | **值：** |
| D1 | 1.在有效期内进行  2A. BACnetSpecialEvents包含calendarEntry：WeekNDay  2B:calendarEntry：WeekNDay：指定WeekOfMonth  2C:calendarEntry：WeekNDay：指定WeekOfMonth值为6  2D.日期是某月最后一周  2E.比其他正在进行的BACnetSpecialEvents有更高的事件优先级 | V1 |
| D2 | 1.在有效期内进行  2A. BACnetSpecialEvents包含calendarEntry：WeekNDay  2B:calendarEntry：WeekNDay：指定WeekOfMonth  2C:calendarEntry：WeekNDay：指定WeekOfMonth值6  2D.日期不与calendarEntry匹配 | 不等于V1 |

测试步骤：

1. VERIFY Present\_Value = (any value other than V1)

2. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D1) | MAKE (the local date and time = D1)

3. WAIT **Schedule Evaluation Fail Time**

4. VERIFY Present\_Value = V1

5. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D2) | MAKE (the local date and time = D2)

6. WAIT **Schedule Evaluation Fail Time**

7. VERIFY Present\_Value = (any value other than V1)

1. 日历录入Week的WeekNDay测试

所需测试：ReadProperty服务执行测试，10.18;TimeSynchronization服务执行测试，10.30。

BACnet参考条款：12.24.8。

目的：验证Exception\_Schedule中与WeekNDay的DayOfWeek匹配的日期能否启用引用时间表对象。

测试方案：TD根据表8-8中标准更改IUT本地日期和时间，检测Present\_Value属性，验证是否发生了预计的写操作。

配置要求：IUT包含一个时间表对象，其Exception\_Schedule包含一个BACnet日历录入，其中WeekNDay录入指定为某周的某天。表8-8给出了测试中使用日期的标准。设置本地日期和时间，使Present\_Value属性值不为V1。如果设备对象有 Protocol\_Revision属性且值大于等于4，则不执行此测试。

**表8-8** 日历录入DayOfWeek的WeekNDay测试日期和值的标准

|  |  |  |
| --- | --- | --- |
| **日期：** | **标准：** | **值：** |
| D1 | 1. 在有效期内进行  2A. BACnetSpecialEvents包含calendarEntry：WeekNDay  2B:calendarEntry：WeekNDay：指定DayOfWeek  2C:日期是某周指定一天  2D.比其他正在进行的BACnetSpecialEvents有更高的事件优先级 | V1 |
| D2 | 1. 在有效期内进行  2A. BACnetSpecialEvents包含calendarEntry：WeekNDay  2B:calendarEntry：WeekNDay：指定DayOfWeek  2C:日期不是某周指定一天 |  |

测试步骤：

1. VERIFY Present\_Value = (any value other than V1)

2. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D1) | MAKE (the local date and time = D1)

3. WAIT **Schedule Evaluation Fail Time**

4. VERIFY Present\_Value = V1

5. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D2) | MAKE (the local date and time = D2)

6. WAIT **Schedule Evaluation Fail Time**

7. VERIFY Present\_Value = (any value other than V1)

1. 事件优先级测试

所需测试：ReadProperty服务执行测试，10.18;TimeSynchronization服务执行测试，10.30。

BACnet参考条款：12.24.8。

目的：验证指定日期相同时，优先级高的BACnet特殊事件优先于优先级低的BACnet特殊事件。

配置要求：IUT配置一个包含两个或多个BACnet特殊事件的时间表对象，这两个特殊事件在同一天激活，有不同的事件优先级值和不同的BACnet时间值。所有BACnet特殊事件都尽量有相同时间但不同值的BACnet时间值。D1表示所有特殊事件都处于有效日期和时间。

测试步骤：

1. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D1) | MAKE (the local date and time = D1)

2. WAIT **Schedule Evaluation Fail Time**

3. VERIFY Present\_Value = (the value corresponding to the special event with the highest eventPriority)

1. BACnetTimeValue列表测试

所需测试：ReadProperty服务执行测试，10.18;TimeSynchronization服务执行测试，10.30。

BACnet参考条款：12.24.8。

目的：验证Special\_Event全部时间表可执行。

测试方案:特殊事件时间表包含多个BACnetTimeValue数据，设置本地日期和时间值为与每个BACnetTimeValue条目匹配的值，检测Present\_Value属性，验证是否发生了预计的写操作。

配置要求：IUT配置一个包含具有两个或更多BACnet时间值条目的BACnet特殊事件的时间表对象，该对象不包含具有更高优先级的BACnet特殊事件。每个BACnetTimeValue条目值应不同。

测试步骤：

1. REPEAT Di = (the times used in the BACnetTimeValue pairs of the special event) DO {

(TRANSMIT TimeSynchronization-Request, ‘Time’ = Di) | MAKE (the local date and time = Di)

WAIT **Schedule Evaluation Fail Time**

VERIFY Present\_Value = (the value corresponding to the special event with the highest eventPriority)

}

1. Weekly\_Schedule和Exception\_Schedule交互测试

所需测试：ReadProperty服务执行测试，10.18; 时间同步服务执行测试，10.30。

BACnet参考条款：12.24.7,12.24.8。

目的：验证同时发生的情况下，Exception\_Schedule优先于BACnet日时间表。

测试方案：IUT配置了适用于同一时间的Weekly\_Schedule和Exception\_Schedule。本地日期和时间更改为Exception\_Schedule在控制状态的时间，读取Present\_Value，验证是否发生了计划的写操作。如果Weekly\_Schedule处于控制状态，将本地日期和时间更改为一个可能导致状态变更的值，读取Present\_Value，验证Exception\_Schedule是否仍在控制状态。

配置要求：IUT配置一个时间表对象，该对象包含适用于相同日期的Weekly\_Schedule和Exception\_Schedule。Exception\_Schedule中的BACnet特殊事件的事件优先级应高于同一时间进行的BACnet特殊事件。设置BACnet时间值，Weekly\_Schedule写入的值应与Exception\_Schedule写入的值不同。令D1表示Exception\_Schedule配置控制和写入值V1时的日期和时间。Weekly\_Schedule中应有至少一个BACnet时间值对，用于指定D1之后、Exception\_Schedule到期前的时间D2。Weekly\_Schedule设为在时间D2写入值V2。

对于Protocol\_Revision为4或高于4的BACnet实现，日期D2值应在D1与NULL值的Exception\_Schedule中的任意数据中选择。

测试步骤：

1. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D1) | MAKE (the local date and time = D1)

2. WAIT **Schedule Evaluation Fail Time**

3. VERIFY Present\_Value = V1

4. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D2) | MAKE (the local date and time = D2)

5. WAIT **Schedule Evaluation Fail Time**

6. VERIFY Present\_Value = V1

1. Exception\_Schedule恢复测试

所需测试：ReadProperty服务执行测试，10.18; ReinitializeDevice服务执行测试，10.27; TimeSynchronization服务执行测试，10.30。

BACnet参考条款：12.24.4,12.24.7,12.24.8,12.24.9。

目的：验证Exception\_Schedule的恢复行为。

测试方案：IUT配置了包含一个Exception\_Schedule的时间表对象，其中，Exception\_Schedule包括除00:00以外的所有BACnetTimeValue数据。将本地日期和时间更改为00:00或Exception\_Schedule中第一条数据值。读取Present\_Value，验证它包含Schedule\_Default值，或用小于4的Protocol\_Revision实现的Vlast。重置IUT，检查Present\_Value，验证它包含Schedule\_Default值，或用小于4的Protocol\_Revision实现的Vlast。

配置要求：IUT配置一个包含Exception\_Schedule的时间表对象，该时间表包括针对特定日期的多个预定写操作，且第一个预定写入计划在相应的Weekly\_Schedule的第一条数据之前实现。在00:00时间内不安排任何写操作，且同一时间内没更高优先级BACnetSpecialEvents发生。在下述测试中，D1表示Exception\_Schedule的00:00有效，并表示BACnetSpecialEvents对中第一个写操作的时间。Vlast表示当天的最后一个BACnetTimeValue对写入的值。如果设备对象有Protocol\_Revision属性且值大于等于4，则不执行此测试。

测试步骤：

1. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D1) | MAKE (the local date and time = D1)

2. WAIT **Schedule Evaluation Fail Time**

3. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 4) THEN

VERIFY Present\_Value = Schedule\_Default

ELSE

VERIFY Present\_Value = Vlast

4. IF (ReinitializeDevice execution is supported) THEN

TRANSMIT ReinitializeDevice-Request,

‘Reinitialized State of Device’ = WARMSTART,

‘Password’ = (any valid password)

RECEIVE BACnet-Simple-ACK-PDU

ELSE

MAKE (the IUT reinitialize)

5. CHECK (Did the IUT perform a WARMSTART reboot?)

6. WAIT **Schedule Evaluation Fail Time**

7. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 4) THEN

VERIFY Present\_Value = Schedule\_Default

ELSE

VERIFY Present\_Value = Vlast

1. Weekly\_Schedule恢复测试

所需测试：ReadProperty服务执行测试，10.18; ReinitializeDevice执行测试，10.27；TimeSynchronization服务执行测试，10.30。

BACnet参考条款：12.24.4,12.24.7,12.24.9。

目的：验证Weekly\_Schedule的恢复行为。

测试方案：IUT配置了包含一个Weekly\_Schedule的时间表对象，其中，Weekly\_Schedule包括除00:00以外的所有BACnet时间值数据。在用于此测试的日期和时间，Exception\_Schedule不包括Weekly\_Schedule。将本地日期和时间更改为00:00或BACnet日时间表中第一条数据值。读取Present\_Value，验证其包含Schedule\_Default值，或用小于4的Protocol\_Revision实现的Vlast。重置IUT，检查Present\_Value，验证它包含Schedule\_Default值，或用小于4的Protocol\_Revision实现的Vlast。

配置要求：IUT配置一个包含Weekly\_Schedule的时间表对象，该时间表包括针对特定日的多个预定写操作。在00:00时间内不安排任何写操作，且同一时间内没更高优先级BACnetSpecialEvents发生。在下述测试中，D1表示Exception\_Schedule的00:00有效，并表示BACnetSpecialEvents对中第一个写操作的时间。Vlast表示当天的最后一个BACnetTimeValue对写入的值。如果设备对象有Protocol\_Revision属性且值大于等于4，则不执行此测试。

测试步骤：

1. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D1) | MAKE (the local date and time = D1)

2. WAIT **Schedule Evaluation Fail Time**

3. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 4) THEN

VERIFY Present\_Value = Schedule\_Default

ELSE

VERIFY Present\_Value = Vlast

4. IF (ReinitializeDevice execution is supported) THEN

TRANSMIT ReinitializeDevice-Request,

‘Reinitialized State of Device’ = WARMSTART,

‘Password’ = (any valid password)

RECEIVE BACnet-Simple-ACK-PDU

ELSE

MAKE (the IUT reinitialize)

5. CHECK (Did the IUT perform a WARMSTART reboot?)

6. WAIT **Schedule Evaluation Fail Time**

7. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 4) THEN

VERIFY Present\_Value = Schedule\_Default

ELSE

VERIFY Present\_Value = Vlast

1. List\_Of\_Object\_Property\_Reference内部测试

所需测试：ReadProperty服务执行测试，10.18；TimeSynchronization服务执行测试，10.30。

BACnet参考条款：12.24.10。

目的：验证时间表对象是否写入IUT中包含的对象和属性。

测试方案：配置时间表对象，写入同一设备中另一个对象的属性。将IUT时钟设置为一对预定写操作之间的时间，验证执行第一个写操作的数据值。将时钟设为第二个，检查时间表对象的Present\_Value，验证写操作执行情况。如果IUT不支持在IUT内写入对象属性，则不执行此测试。

配置要求：IUT配置一个包含List\_Of\_Object\_Property\_References属性的时间表对象，该属性可引用IUT内另一个对象中的至少一个属性。时间表对象配置有Weekly\_Schedule或活动状态的Exception\_Schedule。在Effective\_Period活动状态期间，在BACnetTimeValues列表中至少有两个连续条目值不同，且Exception\_Schedules优先级不能更高。D1表示这两个BACnetTimeValues中的第一个的日期和时间，对应值V1；D2和V2（与V1值不同）表示第二个BACnetTimeValues。时间Dt定义为在D1和D2之间进行。

测试步骤：

1. (TRANSMIT TimeSynchronization-Request, ‘Time’ = Dt) | MAKE (the local date and time = Dt)

2. WAIT **Schedule Evaluation Fail Time**

3. VERIFY Present\_Value = V1

4. VERIFY (value of referenced property in IUT) = V1

5. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D2) | MAKE (the local date and time = D2)

6. WAIT **Schedule Evaluation Fail Time**

7. VERIFY Present\_Value = V2

8. VERIFY (value of referenced property in IUT) = V2

1. List\_Of\_Object\_Property\_Reference外部测试

所需测试：ReadProperty服务执行测试，10.18;TimeSynchronization服务执行测试，10.30。

BACnet参考条款：12.24.10。

目的：验证时间表对象是否写入IUT中包含的对象和属性。

测试方案：配置时间表对象，写入同一设备中另一个对象的属性。将IUT时钟设置为一对预定写操作之间的时间，验证执行第一个写操作的数据值。将时钟设为第二个，检查时间表对象的Present\_Value，验证写操作执行情况。如果IUT不支持在IUT内写入对象属性，则不执行此测试。

配置要求：设置TD，使其支持WriteProperty-Request服务但不支持WritePropertyMultiple-Request请求。IUT配置时间表对象，该对象包含List\_Of\_Object\_Property\_References属性，该属性引用TD包含对象的属性。时间表对象配置有Weekly\_Schedule或活动状态的Exception\_Schedule。在Effective\_Period活动状态期间，在BACnetTimeValues列表中至少有两个连续条目值不同，且Exception\_Schedules优先级不能更高。D1表示这两个BACnet时间值中的第一个的日期和时间，对应值V1；D2和V2（与V1值不同）表示第二个BACnet时间值。时间Dt定义为在D1和D2之间进行。

测试步骤：

1. (TRANSMIT TimeSynchronization-Request, ‘Time’ = Dt) | MAKE (the local date and time = Dt)

2. WAIT **Schedule Evaluation Fail Time**

3. VERIFY Present\_Value = V1

4. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D2) | MAKE (the local date and time = D2)

5. BEFORE **Schedule Evaluation Fail Time**

RECEIVE WriteProperty-Request,

‘Object Identifier’ = (the referenced object in the TD),

‘Property Identifier’ = (the referenced property in the TD),

‘Property Value’ = V2

6. WAIT **Schedule Evaluation Fail Time**

7. VERIFY Present\_Value = V2

注意事项：IUT生成的任何WriteProperty请求都可能有优先级参数。如果含此参数，则参数范围应在不包括6的1-16中。

1. Exception\_Schedule大小变更测试

所需测试：WriteProperty服务执行测试，10.22

BACnet参考条款：Exception\_Schedule，12.24.8

目的：验证当通过写入数组索引更改Exception\_Schedule的大小时，数组的大小会相应更改，且任何新数据都包含一个空的BACnet时间值列表。如果无法更改Exception\_Schedule数组的大小，则不应执行此测试。如果不存在Protocol\_Revision，或者其值小于4，不执行此测试。

配置要求：IUT应配置一个时间表对象，该对象具有可调整大小的Exception\_Schedule数组。

测试方案：设置Exception\_Schedule数组为特定值，分别通过写入其数组大小来增大，通过写入数组来减小，通过写入数组来增大，并通过写入数组大小来减小。

测试步骤：

1. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Schedule object being tested),

‘Property Identifier’ = Exception\_Schedule,

‘Property Value’ = (Exception\_Schedule array of length 2)

2. RECEIVE Simple-ACK-PDU

3. VERIFY Exception\_Schedule = (the value written in step 1)

4. VERIFY Exception\_Schedule = 2, ARRAY INDEX = 0

5. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Schedule object being tested),

‘Property Identifier’ = Exception\_Schedule,

‘Property Array Index’ = 0,

‘Property Value’ = (some value greater than 2)

6. RECEIVE Simple-ACK-PDU

7. VERIFY Exception\_Schedule = (the value written in step 1 with new entries containing empty Lists of BACnetTimeValue))

8. VERIFY Exception\_Schedule = (the value written in step 5), ARRAY INDEX = 0

9. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Schedule object being tested),

‘Property Identifier’ = Exception\_Schedule,

‘Property Value’ = (Exception\_Schedule array of length 2)

10. RECEIVE Simple-ACK-PDU

11. VERIFY Exception\_Schedule = (the value written in step 9)

12. VERIFY Exception\_Schedule = 2, ARRAY INDEX = 0

13. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Schedule object being tested),

‘Property Identifier’ = Exception\_Schedule

‘Property Value’ = (Exception\_Schedule array of length greater than 2)

14. RECEIVE Simple-ACK-PDU

15. VERIFY Exception\_Schedule = (the value written in step 13)

16. VERIFY Exception\_Schedule = (the length of the array written in step 13), ARRAY INDEX = 0

17. TRANSMIT WriteProperty-Request,

‘Object Identifier’ = (the Schedule object being tested),

‘Property Identifier’ = Exception\_Schedule,

‘Property Array Index’ = 0,

‘Property Value’ = 2

18. RECEIVE Simple-ACK-PDU

19. VERIFY Exception\_Schedule = (an array consisting of elements 1&2 from the array written in step 13)

20. VERIFY Exception\_Schedule = 2, ARRAY INDEX = 0

1. 时间表对象Protocol\_Revision 4测试

ANSI/ASHRAE 135-2001附录a中对时间表对象进行了修订，增加了Protocol\_Revision 4。尽管时间表对象的基本结构变化不大，但其操作是完全不同的，原始时间表对象的测试在某些情况下需要修改，而某些情况下需要完全更换，并且需要对一些更改进行新的测试。本节提为使用Protocol\_Revision 4及更高版本的设备的时间表对象运行提供测试。

以下测试针对不要求属性可写或可配置的时间表对象进行。但是若时间表对象可以任何方式配置，则其配置应适配以下测试。若不能以测试所需的方式配置IUT，则省略该测试。如果IUT支持可以在设备外部写入的时间表对象，则在其中一个时间表进行测试。

时间表对象的测试监测计划在指定日期和时间进行的写操作，读取时间表对象的Present\_Value属性进行验证。为合理的时间内执行测试，设备的时钟和日历必须可更改。

对于这些测试，应根据要求预先确定测试的日期和时间。每个测试的“配置要求”部分中的表格给出了在测试中使用的日期/时间值（记为D1，D2等）的标准。符合这些标准的日期/时间值可以从现有的时间表中选择，也可由供应商制定满足这些标准的时间表。

值Vn与每个日期/时间值Dn相关联，并且定义了时间表写操作的时间，与BACnet时间值对Dn的时间值相关联。Vn可以采用任何原始数据类型。

1. 修订版4 Effective\_Period测试

目的：验证时间表对象活动状态时，Effective\_Period是否控制日期的范围。

测试方案：TD根据表8-9中的标准选择两个Date值，其中一个在Effective\_Period之外，另一个对应于Effective\_Period内的已知预定状态。将IUT的本地日期和时间更改为这两个日期之一，监测由List\_Of\_Object\_Property\_References属性参考属性，验证写操作仅在Effective\_Period进行。

配置要求：IUT应配置一个时间表对象，使表8-9中定义的时间段具有唯一的时间表值。设置本地日期和时间，使Present\_Value属性值不为V1。List\_Of\_Object\_Property\_References属性应至少包含时间表对象可改变的IUT中的属性或在另一个设备中可写的属性中的一个。如果无法配置List\_Of\_Object\_Property\_References，则跳过此测试。

**表8-9** Effective\_Period测试日期和值的标准

|  |  |  |
| --- | --- | --- |
| **日期：** | **标准：** | **值：** |
| D1 | 1.在Effective\_Period内进行  2.Date在Weekly\_Schedule或Exception\_Schedule中 | V1 |
| D2 | 1.不在Effective\_Period内进行  2.Date在Weekly\_Schedule或Exception\_Schedule中 | V2不同于V1 |

测试步骤：

1. VERIFY "referenced property" = (any value other than V1)

2. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D1)

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’ = D1)

| MAKE (the local date and time = D1)

3. WAIT **Schedule Evaluation Fail Time**

4. VERIFY "referenced property" = V1

5. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D2)

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’ = D2)

| MAKE (the local date and time = D2)

6. WAIT **Schedule Evaluation Fail Time**

7. VERIFY "referenced property" = V1

1. 修订版4 Weekly\_Schedule属性测试

目的：验证Weekly\_Schedule包含一周中每一天的可区分时间表，并确认日计划一天的整个时间表可以执行。

测试方案：按表8-10所示顺序更改IUT本地日期和时间，监测Present\_Value属性，验证每个独立的预定日期都有写操作进行。

配置要求：IUT必须配置一个时间表对象，该对象包含一个由七个可区分的每日时间表构成的每周时间表，这些时间表应满足表8-10的要求。设置本地日期和时间，使Present\_Value属性值不为V1。如果无法如上配置，则省略该测试。“活动时期”指Exception\_Schedule确定出现在Present\_Value中的值的时间段。

**表8-10** Weekly\_Schedule测试日期和值的标准

|  |  |  |
| --- | --- | --- |
| **日期：** | **标准：** | **值：** |
| D1 | 1. Date在Effective\_Period内进行  2. Date发生在周一  3.Date不在Exception\_Schedule的活动时间段进行 | V1 |
| D2 | 1. Date在Effective\_Period内进行  2. Date发生在周二  3.Date不在Exception\_Schedule的活动时间段进行 | V2，不等于V1 |
| D3 | 1. Date在Effective\_Period内进行  2. Date发生在周三  3.Date不在Exception\_Schedule的活动时间段进行 | V3，不等于V2 |
| D4 | 1. Date在Effective\_Period内进行  2. Date发生在周四  3.Date不在Exception\_Schedule的活动时间段进行 | V4，不等于V3 |
| D5 | 1. Date在Effective\_Period内进行  2. Date发生在周五  3.Date不在Exception\_Schedule的活动时间段进行 | V5，不等于V4 |
| D6 | 1. Date在Effective\_Period内进行  2. Date发生在周六  3. Date不在Exception\_Schedule的活动时间段进行 | V6，不等于V5 |
| D7 | 1.在Effective\_Period内进行  2. Date发生在周日  3.Date不在Exception\_Schedule的活动时间段进行 | V7，不等于V6 |

测试步骤：

1. VERIFY Present\_Value = (any value other than V1)

2. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D1)

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’ = D1)

| MAKE (the local date and time = D1)

3. WAIT **Schedule Evaluation Fail Time**

4. VERIFY Present\_Value = V1

5. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D2)

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’ = D2)

| MAKE (the local date and time = D2)

6. WAIT **Schedule Evaluation Fail Time**

7. VERIFY Present\_Value = V2

8. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D3)

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’ = D3)

| MAKE (the local date and time = D3)

9. WAIT **Schedule Evaluation Fail Time**

10. VERIFY Present\_Value = V3

11. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D4)

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’ = D4)

| MAKE (the local date and time = D4)

12. WAIT **Schedule Evaluation Fail Time**

13. VERIFY Present\_Value = V4

14. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D5)

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’ = D5)

| MAKE (the local date and time = D5)

15. WAIT **Schedule Evaluation Fail Time**

16. VERIFY Present\_Value = Vr

17. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D6)

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’ = D6)

| MAKE (the local date and time = D6)

18. WAIT **Schedule Evaluation Fail Time**

19. VERIFY Present\_Value = V6

20. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D7)

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’ = D7)

| MAKE (the local date and time = D7)

21. WAIT **Schedule Evaluation Fail Time**

22. VERIFY Present\_Value = V7

23. REPEAT X = (the time portion of the BACnetTimeValue entries for one of the daily schedules in Table 7-10) DO {

(TRANSMIT TimeSynchronization-Request, ‘Time’ = X)

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’ = X)

| MAKE (the local date and time = X)

WAIT **Schedule Evaluation Fail Time**

VERIFY Present\_Value = (the scheduled value corresponding to time X)

}

1. 修订版4 Exception\_Schedule属性测试

如果不能使IUT满足本节中一项或多项测试的配置要求，则省略该测试。无法进行配置可能是由于Exception\_Schedule属性不存在或不可变，这是由于Exception\_Schedule中的可用BACnetSpecialEvent记录数量有限或Calendar对象不可用所致。

1. 修订版4日历参考测试

目的：验证在引用的Calendar对象中出现的日期将启用引用Schedule对象。此测试适用于Protocol\_Revision 3及之前，也适用于Protocol\_Revision 4及更高版本的计划对象。

测试方案：TD根据表8-11标准更改IUT本地日期和时间。检测Present\_Value属性值，验证是否发生预计的写操作。

配置要求：IUT配置一个时间表对象，该对象引用日期表非空的日历对象，表8-11给出日期的标准。设置本地日期和时间，使Present\_Value属性值不为V1。

**表8-11** 日历引用日期标准

|  |  |  |
| --- | --- | --- |
| **日期：** | **标准：** | **值：** |
| D1 | 1. 在Effective\_Period内进行  2A. BACnetSpecialEvent通过calendarReference引用日历对象  2B:该日历的Date\_List属性中含此日期  2C. 时间在V1录入之时或之后，在Exception\_Schedule的所有数据前  2D.比其他正在进行的BACnetSpecialEvent有更高的事件优先级 | V1 |
| D2 | 1. 在Effective\_Period内进行  2.不出现在任何BACnetSpecialEvent中  3.与引用时间表对象中的BACnetDailySchedule中的条目日期相同，时间紧随其后。 | V2，不等于V1 |

测试步骤：

1. VERIFY Present\_Value = (any value other than V1)

2. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D1)

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’ = D1)

| MAKE (the local date and time = D1)

3. WAIT **Schedule Evaluation Fail Time**

4. VERIFY Present\_Value = V1

5. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D2)

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’ = D2)

| MAKE (the local date and time = D2)

6. WAIT **Schedule Evaluation Fail Time**

7. VERIFY Present\_Value = V2

1. 修订版4日历录入日期测试

目的：验证Exception\_Schedule中的指定日期能否启用引用时间表对象。

测试方案：TD根据表8-12的标准更改IUT本地日期和时间。检测Present\_Value属性值，验证是否发生预计的写操作。

配置要求：IUT配置一个时间表对象S，该对象的Exception\_Schedule有一个含指定日期的BACnetCalendarEntry，表8-12给出日期的标准。设置本地日期和时间，使Present\_Value属性值不为V1。

**表8-12** 日历输入日期测试日期和值的标准

|  |  |  |
| --- | --- | --- |
| **日期：** | **标准：** | **值：** |
| D1 | 1. 在Effective\_Period内进行  2A. BACnetSpecialEvent包含calendarReference：Date  2B:日期与calendarReference匹配：Date  2C. 时间在V1后，在Exception\_Schedule的所有数据前  2D.比其他正在进行的BACnetSpecialEvent有更高的事件优先级 | V1 |
| D2 | 1. 在Effective\_Period内进行  2.不出现在任何BACnetSpecialEvent中  3.与引用计划对象中的BACnetDailySchedule中的条目与日期相同，时间紧随其后 | V2，不等于V1 |

测试步骤：

1. VERIFY Present\_Value = (any value other than V1)

2. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D1)

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’ = D1)

| MAKE (the local date and time = D1)

3. WAIT **Schedule Evaluation Fail Time**

4. VERIFY S, Present\_Value = V1

5. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D2)

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’ = D2)

| MAKE (the local date and time = D2)

6. WAIT **Schedule Evaluation Fail Time**

7. VERIFY S, Present\_Value = V2

1. 修订版4日历录入DateRange测试

目的：验证出现在Exception\_Schedule的日期范围内的日期将启用引用Schedule对象。

测试方案：TD根据表8-13的标准更改IUT本地日期和时间。检测Present\_Value属性值，验证是否发生预计的写操作。

配置要求：IUT配置一个时间表对象S，其Exception\_Schedule包含带有日期范围的BACnetCalendarEntry，表8-13给出日期的标准。设置本地日期和时间，使Present\_Value属性值不为V1。

**表8-13** 日历录入DateRange测试日期和值的标准

|  |  |  |
| --- | --- | --- |
| **日期：** | **标准：** | **值：** |
| D1 | 1.在Effective\_Period内进行  2A. BACnetSpecialEvent包含calendarReference：DateRange  2B.日期与calendarReference匹配：DateRange  2C.时间在V1后，在Exception\_Schedule的所有数据前  2D.比其他正在进行的BACnetSpecialEvent有更高的事件优先级  2E. DateRange不包含通配符 | V1 |
| D2 | 1.在Effective\_Period内进行  2.不出现在任何BACnetSpecialEvent中  3.与引用时间表对象中的BACnetDailySchedule中的条目的日期相同，时间紧随其后  4. DateRange不包含通配符 | V2，不等于V1 |

测试步骤：

1. VERIFY (S), Present\_Value = any value other than V1

2. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D1)

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’ = D1)

| MAKE (the local date and time = D1)

3. WAIT **Schedule Evaluation Fail Time**

4. VERIFY S, Present\_Value = V1

5. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D2)

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’ = Dr)

| MAKE (the local date and time = D2)

6. WAIT **Schedule Evaluation Fail Time**

7. VERIFY S, Present\_Value = V2

1. 修订版4日历录入WeekNDay Month测试

目的：验证在Exception\_Schedule中指定与WeekNDay的Month字段匹配的日期，可以启用引用Schedule对象。

测试方案：TD根据表8-14的标准更改IUT本地日期和时间。检测Present\_Value属性值，验证是否发生预计的写操作。

配置要求：IUT配置一个时间表对象S，该对象的Exception\_Schedule有一个指定月份（1月-12月）的WeekNDay目录，表8-14给出日期的标准。设置本地日期和时间，使Present\_Value属性值不为V1。

**表8-14** 日历录入WeekNDay Month测试日期和值的标准

|  |  |  |
| --- | --- | --- |
| **日期：** | **标准：** | **值：** |
| D1 | 1. 在Effective\_Period内进行  2A. BACnetSpecialEvent包含calendarReference：WeekNDay  2B. calendarEntry：WeekNDay：Month  2C.日期与calendarReference匹配：WeekNDay  2D.时间在V1后，在Exception\_Schedule的所有数据前  2E.比其他正在进行的BACnetSpecialEvent有更高的事件优先级 | V1 |
| D2 | 1. 在Effective\_Period内进行  2.不出现在任何BACnetSpecialEvent中  3.与引用时间表对象的BACnetDailySchedule日期相同，时间紧随其后 | V2，不等于V1 |

测试步骤：

1. VERIFY (S), Present\_Value = any value other than V1

2. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D1)

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’ = D1)

| MAKE (the local date and time = D1)

3. WAIT **Schedule Evaluation Fail Time**

4. VERIFY S, Present\_Value = V1

5. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D2)

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’ = D2)

| MAKE (the local date and time = Dr)

6. WAIT **Schedule Evaluation Fail Time**

7. VERIFY S, Present\_Value = V2

1. 修订版4日历录入WeekNDay Week Of Month测试

目的：验证与Exception\_Schedule中的WeekNDay的WeekOfMonth字段匹配的日期将启用引用Schedule对象。

测试方案：TD根据表8-15的标准更改IUT本地日期和时间。检测Present\_Value属性值，验证是否发生预计的写操作。

配置要求：IUT配置一个时间表对象S，该对象的Exception\_Schedule有一个指定月份和周的WeekNDay目录，表7-15给出日期的标准。设置本地日期和时间，使Present\_Value属性值不为V1。

**表8-15** 日历录入WeekNDay Week Of Month测试日期和值的标准

|  |  |  |
| --- | --- | --- |
| **日期：** | **标准：** | **值：** |
| D1 | 1. 在Effective\_Period内进行  2A. BACnetSpecialEvent包含calendarReference：WeekNDay  2B. calendarEntry：WeekNDay：指定WeekOfMonth  2C. calendarEntry：WeekNDay：WeekOfMonth范围为1-5  2D.日期与calendarReference匹配：WeekNDay  2E.时间在V1后，在Exception\_Schedule的所有数据前  2F.比其他正在进行的BACnetSpecialEvent有更高的事件优先级 | V1 |
| D2 | 1. 在Effective\_Period内进行  2A. BACnetSpecialEvent包含calendarReference：WeekNDay  2B. calendarEntry：WeekNDay：指定WeekOfMonth  2C. calendarEntry：WeekNDay：WeekOfMonth范围为1-5  2D.不匹配calendarReference：WeekNDay：WeekOfMonth | V2，不等于V1 |

测试步骤：

1. VERIFY (S), Present\_Value = any value other than V1

2. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D1)

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’ = D1)

| MAKE (the local date and time = D1)

3. WAIT **Schedule Evaluation Fail Time**

4. VERIFY S, Present\_Value = Vr

5. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D2)

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’ = D2)

| MAKE (the local date and time = D2)

6. WAIT **Schedule Evaluation Fail Time**

7. VERIFY S, Present\_Value = V2

1. 修订版4日历录入Last WeekNDay测试

目的：验证与Exception\_Schedule中的WeekNDay的WeekOfMonth字段匹配的日期将启用引用Schedule对象。

测试方案：TD根据表8-16的标准更改IUT本地日期和时间。检测Present\_Value属性值，验证是否发生预计的写操作。

配置要求：

IUT必须配置为包含一个Schedule对象S，其Exception\_Schedule包含一个BACnetCalendarEntry，其中的WeekNDay条目指定了该月的最后一周。表7-16中给出了测试中使用的日期标准。应设置本地日期和时间，以使Present\_Value属性的值不是V1。

**表8-16** 日历录入WeekNDay某月最后一周测试日期和值的标准

|  |  |  |
| --- | --- | --- |
| **日期：** | **标准：** | **值：** |
| D1 | 1.在 Effective\_Period,内进行  2A. BACnetSpecialEvent包含calendarReference：WeekNDay  2B. calendarEntry：WeekNDay指定WeekOfMonth  2C. calendarEntry：WeekNDay：WeekOfMonth值为6  2D.日期是某月的最后一周  2E.WeekNDay：Month与指定月份匹配  2F.WeekNDay：dayOfWeek与指定某天匹配  2G.时间在V1后，在Exception\_Schedule的所有数据前  2H.比其他正在进行的BACnetSpecialEvent有更高的事件优先级 | V1 |
| D2 | 1.在 Effective\_Period,内进行  2A. BACnetSpecialEvent包含calendarReference：WeekNDay  2B. calendarEntry：WeekNDay指定WeekOfMonth  2C. calendarEntry：WeekNDay：WeekOfMonth值为6  2D.日期不是某月的最后一周 | V2，不等于V1 |

测试步骤：

1. VERIFY (S), Present\_Value = any value other than V1

2. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D1) |

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’ = D1)

MAKE (the local date and time = D1)

3. WAIT **Schedule Evaluation Fail Time**

4. VERIFY S, Present\_Value = V1

5. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D2) |

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’ = D2)

MAKE (the local date and time = D2)

6. WAIT **Schedule Evaluation Fail Time**

7. VERIFY S, Present\_Value = V2

1. 写数据类型测试

以下测试验证时间表是否能正确写入其支持的数据类型。

1. 内部写数据类型测试（非NULL值）

BACnet参考条款：12.24,12.24.10

目的：验证IUT时间表对象对正在测试的非NULL的数据类型的设备写入属性。

测试方案：TD根据表8-17中的标准选择两个日期/时间值D1和D2，D1应晚于当前时间很多，当时间变成D1时，进行评估。将时间从D1设置为D2（晚于D1），时间变成D2时进行评估，从而使其写入值V2。可以基于时间表对象S的现有配置来选择这些值，或用这些值来配置时间表对象。

配置要求：IUT配置一个时间表对象S，使表7-17中定义的时间段计划值唯一。时间表对象应配置List\_Of\_Object\_Property\_References，至少引用设备内一个可写属性。如果IUT不能按上述要求配置，则省略该测试。其他属性在数据类型和值上应保持一致，并应允许执行此测试。

**表8-17** 测试日期时间标准

|  |  |
| --- | --- |
| **日期和时间：** | **值：** |
| D1 | V1 |
| D2 | V2（不等于V1） |

测试步骤：

1. (TRANSMIT TimeSynchronization-Request,’Time’ = D1)

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’=D1)

| MAKE (the local date and time = D1)

2. WAIT(**Schedule\_Evaluation\_Fail\_Time**)

3. VERIFY S, Present\_Value = V1

4. REPEAT P = (writable property in List\_Of\_Property\_References)

VERIFY P = V1

5. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D2)

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’=D1)

| MAKE (the local date and time = D2)

6. WAIT(**Schedule\_Evaluation\_Fail\_Time**)

7. VERIFY S, Present\_Value = V2

8. REPEAT P = (writable property in List\_Of\_Property\_References)

VERIFY P = V2

注意事项：此测试中可写意味着时间表对象能够修改属性，但并不表明该属性可通过BACnet服务进行修改。

1. 内部写入数据类型测试（NULL值，Priority\_Array）

目的：验证时间表对象可在同一设备中通过Present\_Value将NULL写入Priority\_Array。

所需测试：TimeSynchronization服务执行测试，10.30，UTCTimeSynchronization服务执行测试，10.31

BACnet参考条款：12.24,12.24.9

测试方案：TD根据表8-18标准选择两个日期/时间值D1和D2，以便当时间更改为D1时，D1与当前时间有足够的差异进行时间表评估，并时间从D1设置为D2时进行评估，从而使其写入值V2。可以基于时间表对象S的现有配置来选择这些值，或用这些值来配置时间表对象。V1或V2，数据类型为NULL，但不是两者同时为NULL。这些值将写入Present\_Value属性，其优先级由时间表对象的写入优先级属性X指定。对于协议版本4或更高版本的设备，设置Schedule\_Default为NULL，配置时间表使在时间D1或D2时间表采用Schedule\_Default的值，但D1、D2不能同时采用。

配置要求：IUT配置一个时间表对象S，使表8-18中定义的时间段计划值唯一。时间表对象应配置为List\_Of\_Object\_Property\_References，至少引用含Priority\_Array属性PA的设备内的一个Present\_Value属性P。如果IUT不能按上述要求配置，则省略该测试。其他属性在数据类型和值上应保持一致，并应允许执行此测试。

**表8-18** 测试日期时间标准

|  |  |
| --- | --- |
| **日期和时间：** | **值：** |
| D1 | V1 |
| D2 | V2（不等于V1） |

测试步骤：

1. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D1)

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’=D1)

| MAKE (the local date and time = D1)

2. WAIT(**Schedule\_Evaluation\_Fail\_Time**)

3. VERIFY S, Present\_Value = V1

4. VERIFY PA[X] = V1

5. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D2)

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’=D2)

| MAKE (the local date and time = D2)

6. WAIT(**Schedule\_Evaluation\_Fail\_Time**)

7. VERIFY S, Present\_Value = V2

8. VERIFY PA[X] = V2

1. 内部写数据类型测试（非NULL值）

所需测试：TimeSynchronization服务执行测试，10.26，UTCTimeSynchronization服务执行测试，10.31

BACnet参考条款：12.24,12.24.10

目的：验证Schedule对象是否使用外部写入操作要求和声明的所有数据类型写入其他设备中的属性。 如果IUT支持在支持的数据类型上具有差异的Schedule对象，则应在每种类型的至少一个实例上执行此测试。

测试方案：TD根据表8-19标准选择两个日期/时间值D1和D2，D1应晚于当前时间，当时间变成D1时，进行评估。将时间从D1设置为D2（晚于D1），时间变成D2时进行评估，从而使其写入值V2。可以基于时间表对象S的现有配置来选择这些值，或用这些值来配置时间表对象。

配置要求：TD应支持WriteProperty-Request服务，不支持其设备对象的Protocol\_Services\_Supported属性中的WritePropertyMultiple-Request。IUT配置时间表对象S，使表8-19中定义的时间段计划值唯一。时间表对象应配置为List\_Of\_Object\_Property\_References，至少引用设备内一个可写属性。如果IUT不能按上述要求配置，则省略该测试。其他属性在数据类型和值上应保持一致，并应允许执行此测试。

**表8-19** 测试日期时间标准

|  |  |
| --- | --- |
| **日期和时间：** | **值：** |
| D1 | V1 |
| D2 | V2（不等于V1） |

测试步骤：

1. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D1)

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’=D1)

| MAKE (the local date and time = D1)

2. BEFORE **Schedule\_Evaluation\_Fail\_Time**

REPEAT X = (every reference to the TD in List\_Of\_Object\_Property\_References) DO {

RECEIVE WriteProperty-Request,

‘Object Identifier’ = (the object identifier of X),

‘Property Identifier’ =(the property of X),

‘Property Value’ = V1

‘Priority’ = (the value of the Schedule object’s Priority\_For\_Writing property)

TRANSMIT BACnet-SimpleACK-PDU

}

3. VERIFY S, Present\_Value = V1

4. (TRANSMIT TimeSynchronization-Request, ‘ Time’ = D2)

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’=D2)

| MAKE (the local date and time = D2)

5. BEFORE **Schedule\_Evaluation\_Fail\_Time**

REPEAT X = (every reference to the TD in List\_Of\_Object\_Property\_References) DO {

RECEIVE WriteProperty-Request,

‘Object Identifier’ = (the object identifier of X),

‘Property Identifier’ =(the property of X),

‘Property Value’ = V2,

‘Priority’ = (the value of the Schedule object’s Priority\_For\_Writing property)

TRANSMIT BACnet-SimpleACK-PDU

}

6. VERIFY S, Present\_Value = V2

注意事项：如果Schedule在其Priority\_For\_Writing属性中配置的值为16，则可能会忽略WriteProperty-Request的Priority参数。无论IUT在步骤2和5中生成WriteProperty-Requests的顺序如何，测试都应通过。

1. 外部写入数据类型测试（NULL值，Priority\_Array）

所需测试：TimeSynchronization服务执行测试，10.26，UTCTimeSynchronization服务执行测试，10.31

BACnet参考条款：12.24,12.24.9

目的：验证时间表对象可在其他设备中通过Present\_Value将NULL写入Priority\_Array。

测试方案：TD根据表8-20中的标准选择两个日期/时间值D1和D2，当时间更改为D1时，D1与当前时间有足够的差异进行时间表评估，且将时间从D1设置为D2会导致评估，从而使其写入值V2。可以根据Schedule对象的现有配置来选择这些值，或者可以为Schedule对象配置这些值，V1或V2，数据类型为NULL，但不是两者同时为NULL。这些值将以Schedule对象的Priority\_For\_Writing属性指定的优先级写入Present\_Value属性。对于协议版本4或更高版本的设备，Schedule\_Default应设置为NULL，配置时间表使在时间D1或D2时间表采用Schedule\_Default的值，但D1、D2不能同时采用。

配置要求：TD必须配置为支持WriteProperty-Request服务，但不支持其Device对象的Protocol\_Services\_Supported属性中的WritePropertyMultiple-Request。IUT必须配置有一个Schedule对象S，以便可以使用唯一的预定值来配置表8-20中定义的时间段。时间表对象应配置一个不为16的Priority\_For\_Writing值，并配置一个List\_Of\_Object\_Property\_References，其中对有Priority\_Array属性的TD中的对象中包括对Present\_Value属性的至少一个引用。

**表8-20** 测试日期时间标准

|  |  |
| --- | --- |
| **日期和时间：** | **值：** |
| D1 | V1 |
| D2 | V2（不等于V1） |

测试步骤：

1. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D1)

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’=D1)

| MAKE (the local date and time = D1)

2. BEFORE **Schedule\_Evaluation\_Fail\_Time**

REPEAT X = (every reference to the TD in List\_Of\_Object\_Property\_References) DO {

RECEIVE WriteProperty-Request,

‘Object Identifier’ = (the object identifier of X),

‘Property Identifier’ = (the property of X),

‘Property Value’ = V1,

‘Priority’ = (the value of the Schedule object’s Priority\_For\_Writing property)

TRANSMIT BACnet-SimpleACK-PDU

}

3. VERIFY S, Present\_Value = V1

4. (TRANSMIT TimeSynchronization-Request, ‘Time’ = D2)

| (TRANSMIT UTCTimeSynchronization-Request, ‘Time’=D2)

| MAKE (the local date and time = D2)

5. BEFORE **Schedule\_Evaluation\_Fail\_Time**

REPEAT X = (every reference to the TD in List\_Of\_Object\_Property\_References) DO {

RECEIVE WriteProperty-Request,

‘Object Identifier’ = (the object identifier of X),

‘Property Identifier’ =(the property of X),

‘Property Value’ = V2,

‘Priority’ = (the value of the Schedule object’s Priority\_For\_Writing property)

TRANSMIT BACnet-SimpleACK-PDU

}

6. VERIFY S, Present\_Value = V2

注意事项：如果Schedule在其Priority\_For\_Writing属性中配置的值为16，则可能会忽略WriteProperty-Request的Priority参数。无论IUT在步骤2和5中生成WriteProperty-Requests的顺序如何，测试都通过。

1. 趋势日志对象测试

趋势日志对象只有几个属性要求可写或可配置。趋势日志对象的配置应尽可能符合以下测试。如果不能以测试所需的方式去配置IUT，则省略此测试。

趋势日志对象的测试关注其日志缓冲区的记录（时间，值）以及自启动或上一次通知以来收集到预定数量的记录时的通知发布。

1. Log\_Enable测试

所需测试：ReadProperty服务执行测试，10.18;WriteProperty服务执行测试，10.22。

BACnet参考条款：12.25.5。

目的：验证Log\_Enable属性启用和禁用趋势日志对象记录的数据。

测试方案：趋势日志设置为通过两种方式（轮询和COV订阅）获取数据。启用Log\_Enable，确认日志缓冲区中有一个或多个记录的集合；然后禁用Log\_Enable，确认无收集记录。

测试使用的COV增量是趋势日志的Client\_COV\_Increment属性或被监测对象的COV\_Increment属性，具体取决于要测试的趋势日志对象的配置。

配置要求：Start\_Time（若存在）设置为测试开始之前的日期和时间。Stop\_Time（若存在）设置为测试完成后的时间。Stop\_When\_Full若可配置，则设为FALSE。

测试步骤：

1. READ I = Log\_Interval

2. WRITE Log\_Enable = FALSE

3. WRITE Record\_Count = 0

4. WAIT **Internal Processing Fail Time**

5. WRITE Log\_Enable = TRUE

6. READ X = Total\_Record\_Count

7. IF (I = 0) THEN

MAKE (monitored value change more thanthe COV increment)

ELSE

WAIT (I)

8. WAIT (**Notification Fail Time** + **Internal Processing Fail Time**)

9. VERIFY Total\_Record\_Count > X

10. WRITE Log\_Enable = FALSE

11. READ Y = Total\_Record\_Count

12. IF (I = 0) THEN

MAKE (monitored value change more than the COV increment)

ELSE

WAIT (I)

13. WAIT (**Notification Fail Time** + **Internal Processing Fail Time**)

14. VERIFY Total\_Record\_Count = Y

1. Start\_Time测试

所需测试：ReadProperty服务执行测试，10.18;WriteProperty服务执行测试，10.22。

BACnet参考条款：12.25.6。

目的：验证是否在Start\_Time指定的时间启用日志记录。

测试方案：趋势日志设置为通过两种方式（轮询和COV订阅）获取数据。在Start\_Time中指定的时间之前的某个时间开始测试，确认未收集记录。然后在Start\_Time指定的时间之后的某个时间开始测试，确认收集记录。

配置要求：Start\_Time配置一个日期和时间满足步骤1到6将在该时间之前结束。Stop\_Time（如果存在）应配置新的日期和时间使其在测试结束后发生。Stop\_When\_Full（如果可配置）应设置为FALSE； Log\_Enable设置为TRUE。

测试步骤：

1. WRITE Record\_Count = 0

2. WAIT **Internal Processing Fail Time**

3. TRANSMIT ReadProperty-Request,

‘Object Identifier’ = (the object being tested),

‘Property Identifier’ = Total\_Record\_Count

4. RECEIVE ReadProperty-ACK,

‘Object Identifier’ = (the object being tested),

‘Property Identifier’ = Total\_Record\_Count

‘Property Value’ = (any valid value, X)

5. IF (COV subscription in use) THEN

MAKE (monitored value change more than Client\_COV\_Increment)

ELSE

WAIT (Log\_Interval)

6. WAIT (**Notification Fail Time** + **Internal Processing Fail Time**)

7. VERIFY Total\_Record\_Count = (value X returned in step 4)

8. WHILE (IUT clock is earlier than Start\_Time) DO {

VERIFY Total\_Record\_Count = (value X returned in step 4)

}

9. WAIT (**Notification Fail Time** + **Internal Processing Fail Time**)

10. IF (COV subscription in use) THEN

MAKE (monitored value change more than Client\_COV\_Increment)

ELSE

WAIT (Log\_Interval)

11. WAIT (**Notification Fail Time** + **Internal Processing Fail Time**)

12. VERIFY Total\_Record\_Count > (value X returned in step 4)

1. Stop\_Time测试

所需测试：ReadProperty服务执行测试，10.18;WriteProperty服务执行测试，10.22。

BACnet参考条款：12.25.7。

目的：验证在Stop\_Time指定的时间禁用日志记录。

测试方案：趋势日志设置为通过两种方式（轮询和COV订阅）获取数据。在Start\_Time中指定的时间之前的某个时间开始测试，确认收集记录。在Stop\_Time指定的时间之后则不再收集记录。

配置要求：设置Stop\_Time日期和时间，使其在此时间前完成步骤1到9。设置Start\_Time（若存在）为测试开始前的日期和时间。设置Stop\_When\_Full为FALSE（若可配置）。

测试步骤：

1. WRITE Log\_Enable = FALSE

2. WAIT **Internal Processing Fail Time**

3. WRITE Record\_Count = 0

4. WRITE Log\_Enable = TRUE

5. READ X = Record\_Count

6. WAIT **Internal Processing Fail Time**

7. IF (COV subscription in use) THEN

MAKE (monitored value change more than Client\_COV\_Increment)

ELSE

WAIT (Log\_Interval)

8. WAIT (**Notification Fail Time** + **Internal Processing Fail Time**)

9. VERIFY Total\_Record\_Count > X

10. WHILE (IUT clock is earlier than Stop\_Time) DO {}

11. WAIT (**Notification Fail Time** + **Internal Processing Fail Time**)

12. READ X = Total\_Record\_Count

13. IF (COV subscription in use) THEN

MAKE (monitored value change more than Client\_COV\_Increment)

ELSE

WAIT (Log\_Interval)

14. WAIT (**Notification Fail Time** + **Internal Processing Fail Time**)

15. VERIFY Total\_Record\_Count = X

1. Log\_Interval测试

所需测试：ReadProperty服务执行测试，10.18;WriteProperty服务执行测试，10.22。

BACnet参考条款：12.25.9。

目的：验证记录周期是否由Log\_Interval控制。

测试方案：设置趋势日志为通过轮询获取数据。轮询由Log\_Interval定义的两个不同的间隔完成，每个速率获得大约10个记录。检查记录的时间戳，验证轮询率。

配置要求：设置Start\_Time（若存在）为测试开始前的日期和时间；设置Stop\_Time（若存在）为可能的最新日期和时间，使其在测试结束后进行。设置Stop\_When\_Full为FALSE（若可配置）。设置Log\_Enable为TRUE。根据供应商为此属性指定的范围和分辨率，为Log\_Interval选择非零值。

测试步骤：

1. WRITE Log\_Interval = (some non-zero value)

2. WRITE Record\_Count = 0

3. WAIT (**Internal Processing Fail Time** + 10\* Log\_Interval hundredths-seconds)

4. VERIFY (Log\_Buffer record timestamp intervals, on average, are as written in step 1)

5. WRITE Log\_Interval = (a non-zero value different from the one written in step 1)

6. WRITE Record\_Count = 0

7. WAIT (**Internal Processing Fail Time** + 10\* Log\_Interval hundredths-seconds)

8. VERIFY (Log\_Buffer record timestamp intervals, on average, are as written in step 5)

1. COV\_Resubscription\_Interval测试

所需测试：确认报告订阅，9.10.1。

BACnet参考条款：12.25.10。

目的：验证通过COV通知获取数据的趋势日志将按COV\_Resubscription\_Interval设置的时间间隔重新发布其订阅。

测试方案：设置趋势日志使其通过COV报告从TD获取数据，TD验证重新订阅间隔。

配置要求：设置Start\_Time（若存在）为测试开始前的日期和时间；设置Stop\_Time（若存在）为可能的最新日期和时间，使其在测试结束后进行。设置Stop\_When\_Full为FALSE（若可配置）；设置Log\_Enable为TRUE。应根据供应商为此属性指定的范围和分辨率，为Log\_Interval选择非零值。

测试步骤：

1. IF (the IUT uses SubscribeCOV) THEN

RECEIVE SubscribeCOV-Request,

‘Subscriber Process Identifier’ = (any value),

‘Monitored Object Identifier’ = (the object to be monitored),

‘Issue Confirmed Notifications’ = (TRUE),

‘Lifetime’ = (any value >= COV\_Resubscription\_Interval)

ELSE

RECEIVE SubscribeCOVProperty-Request,

‘Subscriber Process Identifier’ = (any value),

‘Monitored Object Identifier’ = (the object to be monitored),

‘Issue Confirmed Notifications’ = (TRUE),

‘Lifetime’ = (any value >= COV\_Resubscription\_Interval),

‘Monitored Property Identifier’ = (the property to be monitored),

‘COV Increment’ = (Client\_COV\_Increment -- optional)

2. TRANSMIT BACnet-SimpleACK-PDU

3. TRANSMIT ConfirmedCOVNotification-Request,

‘Subscriber Process Identifier’ = (corresponding value in step 1),

‘Initiating Device Identifier’ = (Device object identifier of the TD),

‘Monitored Object Identifier’ = (corresponding value in step 1),

‘Issue Confirmed Notifications’ = (corresponding value in step 1),

‘Time Remaining’ = (any value <= the Lifetime from step 1),

‘List of Values’ = (appropriate BACnetPropertyValue(s))

4. RECEIVE BACnet-SimpleACK-PDU

5. BEFORE (the lesser of COV\_Resubscription\_Interval + **Re-subscription Interval Tolerance** and LifeTime from step 1)

IF (the IUT uses SubscribeCOV)

RECEIVE SubscribeCOV-Request,

‘Subscriber Process Identifier’ = (corresponding value in step 1),

‘Monitored Object Identifier’ = (corresponding value in step 1),

‘Issue Confirmed Notifications’ = (TRUE),

‘Lifetime’ = (any value >= COV\_Resubscription\_Interval)

ELSE

RECEIVE SubscribeCOVProperty-Request,

‘Subscriber Process Identifier’ = (corresponding value in step 1),

‘Monitored Object Identifier’ = (corresponding value in step 1),

‘Issue Confirmed Notifications’ = (TRUE),

‘Lifetime’ = (any value >= COV\_Resubscription\_Interval),

‘Monitored Property Identifier’ = (corresponding value in step 1),

‘COV Increment’ = (corresponding value in step 1)

6. TRANSMIT BACnet-SimpleACK-PDU

7. TRANSMIT ConfirmedCOVNotification-Request,

‘Subscriber Process Identifier’ = (corresponding value in step 1),

‘Initiating Device Identifier’ = (Device object identifier of the TD),

‘Monitored Object Identifier’ = (corresponding value in step 1),

‘Issue Confirmed Notifications’ = (corresponding value in step 1),

‘Time Remaining’ = (any value <= the Lifetime from step 5),

‘List of Values’ = (appropriate BACnetPropertyValue(s))

8. RECEIVE BACnet-SimpleACK-PDU

9. WAIT (COV\_Resubscription\_Interval - **Re-subscription Interval Tolerance**)

10. BEFORE (2 \* **Re-subscription Interval Tolerance**)

IF (the IUT uses SubscribeCOV)

RECEIVE SubscribeCOV-Request,

‘Subscriber Process Identifier’ = (corresponding value in step 1),

‘Monitored Object Identifier’ = (corresponding value in step 1),

‘Issue Confirmed Notifications’ = (TRUE),

‘Lifetime’ = (corresponding value in step 1)

ELSE

RECEIVE SubscribeCOVProperty-Request,

‘Subscriber Process Identifier’ = (corresponding value in step 1),

‘Monitored Object Identifier’ = (corresponding value in step 1),

‘Issue Confirmed Notifications’ = (TRUE),

‘Lifetime’ = (corresponding value in step 1),

‘Monitored Property Identifier’ = (corresponding value in step 1),

‘COV Increment’ = (corresponding value in step 1)

11. TRANSMIT BACnet-SimpleACK-PDU

结果：如果SubscribeCOV的Lifetime参数小于COV\_Resubscription\_Interval + 再订阅间隔容限时，即使该时间窗小于测试定义的时间窗，IUT也应在Lifetime秒内发送后续的SubscribeCOV请求。如果IUT不符合，则不通过测试。

1. Stop\_When\_Full测试

在Stop\_When\_Full上执行两个测试。第一个测试在Stop\_When\_Full设置为TRUE时执行，第二个是在Stop\_When\_Full设为FALSE时执行。

1. Stop\_When\_Full TRUE测试

所需测试：ReadProperty服务执行测试，10.18; WriteProperty服务执行测试，10.22。

BACnet参考条款：12.25.12。

目的：验证Stop\_When\_Full设为TRUE时是否表示趋势日志在其Log\_Buffer获取Buffer\_Size数据项时停止收集数据。

测试方案：设置趋势日志使其可通过任何方式获取数据。收集到超过Buffer\_Size条记录时停止收集数据，验证Log\_Enable为FALSE。

配置要求：设置Start\_Time（若存在）为测试开始前的日期和时间；设置Stop\_Time（若存在）为可能的最新日期和时间，使其在测试结束后进行。设置Stop\_When\_Full为FALSE（若可配置）；设置Log\_Enable为TRUE。

测试方案：

1. WRITE Record\_Count = 0

2. WAIT **Internal Processing Fail Time**

3. TRANSMIT ReadProperty-Request,

‘Object Identifier’ = (the object being tested),

‘Property Identifier’ = Total\_Record\_Count,

4. RECEIVE ReadProperty-ACK,

‘Object Identifier’ = (the object being tested),

‘Property Identifier’ = Total\_Record\_Count,

‘Property Value’ = (any valid value, X)

5. WRITE Log\_Enable = TRUE

6. WHILE ( (Total\_Record\_Count – (value X returned in step 4)) modulo 232 < Buffer\_Size ) DO { }

7. WAIT **Internal Processing Fail Time**

8. VERIFY Log\_Enable = FALSE

1. Stop\_When\_Full FALSE测试

所需测试：ReadProperty服务执行测试，10.18; WriteProperty服务执行测试，10.22。

BACnet参考条款：12.25.12。

目的：验证Stop\_When\_Full设为FALSE时是否表示趋势日志在其Log\_Buffer获取Buffer\_Size数据项后继续收集数据。

测试方案：设置趋势日志使其可通过任何方式获取数据。直到收集了不止Buffer\_Size个记录并且验证Log\_Enable为TRUE为止，才收集数据。收集到超过Buffer\_Size条记录时收集数据，验证Log\_Enable为TRUE。

配置要求：设置Start\_Time（若存在）为测试开始前的日期和时间；设置Stop\_Time（若存在）为可能的最新日期和时间，使其在测试结束后进行。设置Stop\_When\_Full为FALSE（若可配置）；设置Log\_Enable为FALSE。

测试步骤：

1. WRITE Record\_Count = 0

2. WAIT **Internal Processing Fail Time**

3. TRANSMIT ReadProperty-Request,

‘Object Identifier’ = (the object being tested),

‘Property Identifier’ = Total\_Record\_Count

4. RECEIVE ReadProperty-ACK,

‘Object Identifier’ = (the object being tested),

‘Property Identifier’ = Total\_Record\_Count,

‘Property Value’ = (any valid value, X)

5. WRITE Log\_Enable = TRUE

6. WHILE ( (Total\_Record\_Count – (value X returned in step 4)) modulo 232 < (Buffer\_Size+1) ) DO { }

7. WAIT **Internal Processing Fail Time**

8. VERIFY Log\_Enable = TRUE

1. Buffer\_Size测试

所需测试：ReadProperty服务执行测试，10.18; WriteProperty服务执行测试，10.22。

BACnet参考条款：12.25.13。

目的：验证Buffer\_Size是否正确表示Log\_Buffer中存储的记录数。

测试方案：设置趋势日志使其可通过任意方式获取数据。收集数据，直到至少已收集Buffer\_Size记录，读取Log\_Buffer，验证存在Buffer\_Size记录。

配置要求：设置Start\_Time（若存在）为测试开始前的日期和时间；设置Stop\_Time（若存在）为可能的最新日期和时间，使其在测试结束后进行；设置Log\_Enable为TRUE。

测试步骤：

1. WHILE ( Record\_Count < Buffer\_Size ) DO { }

2. WRITE Log\_Enable = FALSE

3. WAIT **Internal Processing Fail Time**

4. CHECK ( that Log\_Buffer has Buffer\_Size discrete records)

1. Record\_Count测试

所需测试：ReadProperty服务执行测试，10.18; WriteProperty服务执行测试，10.22。

BACnet参考条款：12.25.15。

目的：验证Record\_Count属性表示Log\_Buffer中存储的记录数。

测试方案：设置趋势日志使其可通过任意方式获取数据。收集Buffer\_Size记录，读取日志缓冲区，验证存在Buffer\_Size记录。

测试方案：设置趋势日志使其可通过任意方式获取数据。设置Record\_Count为零，读取Log\_Buffer，验证没有记录。继续收集数据直到Record\_Count为Buffer\_Size / 2，暂停收集并读取Log\_Buffer，检测Record\_Count。再恢复收集直到读取了Buffer\_Size条记录，暂停收集并读取Log\_Buffer，再次验证Record\_Count。

配置要求：设置Start\_Time（若存在）为测试开始前的日期和时间；设置Stop\_Time（若存在）为可能的最新日期和时间，使其在测试结束后进行；设置Log\_Enable为FALSE。

测试步骤：

1. WRITE Record\_Count = 0

2. WAIT **Internal Processing Fail Time**

3. CHECK ( that Log\_Buffer has no records )

4. WRITE Log\_Enable = TRUE

5. WHILE ( Record\_Count < Buffer\_Size/2 ) DO { }

6. WRITE Log\_Enable = FALSE

7. WAIT **Internal Processing Fail Time**

8. VERIFY ( that Log\_Buffer has the number of records indicated by Record\_Count )

9. WRITE Log\_Enable = TRUE

10. WHILE ( Record\_Count < Buffer\_Size ) DO { }

11. WRITE Log\_Enable = FALSE

12. WAIT **Internal Processing Fail Time**

13. VERIFY ( that Log\_Buffer has the number of records indicated by Record\_Count )

1. Total\_Record\_Count测试

所需测试：ReadProperty服务执行测试，10.18; WriteProperty服务执行测试，10.22。

BACnet参考条款：12.25.16。

目的：验证添加到Log\_Buffer的每个记录的Total\_Record\_Count属性增量，即使在添加了Buffer\_Size记录之后也是如此。（注：根据BACnet第12.25.16条的要求测试值从232-1到1是不合理的；即使每隔100秒收集一次记录，也可能需要497天才能完成测试。）

测试方案：趋势日志配置为通过任何方式获取数据。设置Record\_Count为零，读取Total\_Record\_Count；继续收集数据至Record\_Count更改，收集停止并检查Total\_Record\_Count是否已按Record\_Count递增。若无法设置IUT配置，使得TD在收集Buffer\_Size记录之前就停止收集，则不执行此测试。

配置要求：设置Start\_Time（若存在）为测试开始前的日期和时间；设置Stop\_Time（若存在）为可能的最新日期和时间，使其在测试结束后进行；设置Log\_Enable为FALSE。

测试步骤：

1. WRITE Record\_Count = 0

2. WAIT **Internal Processing Fail Time**

3. READ X = Total\_Record\_Count

4. READ Y = Record\_Count

5. WRITE Log\_Enable = TRUE

6. WHILE ( Record\_Count = Y + 1 ) DO { }

7. WRITE Log\_Enable = FALSE

8. WAIT **Internal Processing Fail Time**

9. IF (Total\_Record\_Count - X != Record\_Count - Y) THEN

ERROR “Total\_Record\_Count has incorrect value.”

1. Notification\_Threshold测试

所需测试：ReadProperty服务执行测试，10.18; WriteProperty服务执行测试，10.22。

BACnet参考条款：12.25.17。

目的：验证Notification\_Threshold属性是否反映了自上一个通知以来或自日志记录开始以来收集的发送Buffer\_Ready通知的记录数。

测试方案：设置趋势日志为可通过任意方式获取数据。Record\_Count设置为零。继续收集数据至收到报告，停止收集并检查Record\_Count的值。接收到第二个报告时恢复收集，当再次停止收集时验证Record\_Count。若无法设置IUT配置使TD能够在发出通知后收集另一条记录之前停止收集，则不执行此测试。

配置要求：设置Start\_Time（若存在）为测试开始前的日期和时间；设置Stop\_Time（若存在）为可能的最新日期和时间，使其在测试结束后进行；设置Log\_Enable为FALSE。

测试步骤：

1. WRITE Record\_Count = 0

2. WAIT **Internal Processing Fail Time**

3. TRANSMIT ReadProperty-Request,

‘Object Identifier’ = (the object being tested),

‘Property Identifier’ = Total\_Record\_Count

4. RECEIVE ReadProperty-ACK,

‘Object Identifier’ = (the object being tested),

‘Property Identifier’ = Total\_Record\_Count

‘Property Value’ = (any valid value, X)

5. WRITE Log\_Enable = TRUE

6. MAKE ( Trend Log object collect number of records specified by Notification\_Threshold)

7. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the Trend Log object being tested),

‘Time Stamp’ = (any appropriate BACnetTimeStamp value),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-NORMAL transition),

‘Event Type’ = BUFFER\_READY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = NORMAL,

‘Event Values’ = (BACnetObjectIdentifier of the IUT’s Device object),

(BACnetObjectIdentifier of the Trend Log object),

(any BACnetDateTime),

(current local BACnetDateTime)

8. TRANSMIT BACnet-SimpleACK-PDU

9. WRITE Log\_Enable = FALSE

10. IF ( Total\_Record\_Count – (value read in step 4) != Notification\_Threshold ) THEN

ERROR “Notification\_Threshold value is incorrect.”

11. WRITE Log\_Enable = TRUE

12. MAKE ( Trend Log object collect number of records specified by Notification\_Threshold)

13. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the Trend Log object being tested),

‘Time Stamp’ = (any appropriate BACnetTimeStamp value),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-NORMAL transition),

‘Event Type’ = BUFFER\_READY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = NORMAL,

‘Event Values’ = (BACnetObjectIdentifier of the IUT’s Device object),

(BACnetObjectIdentifier of the Trend Log object),

(BACnetDateTime sent in step 7),

(current local BACnetDateTime)

14. TRANSMIT BACnet-SimpleACK-PDU

15. WRITE Log\_Enable = FALSE

16. IF ( Total\_Record\_Count – (value X returned in step 4) != 2 \* Notification\_Threshold ) THEN

ERROR “Notification\_Threshold value is incorrect.”

1. 报告时间测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet-2001参考条款：12.23.19,12.23.20和12.23.26。

目的：验证Previous\_Notify\_Time和Current\_Notify\_Time参数反映最近报告发送的值。仅当设备对象中存在Protocol\_Revision属性且值为1或2时，才执行此测试。

测试方案：设置趋势日志为可通过任意方式获取数据，Record\_Count设置为零。继续收集数据至收到两个报告，停止收集并检查Previous\_Notify\_Time和Current\_Notify\_Time的值。

配置要求：设置Start\_Time（若存在）为测试开始前的日期和时间；设置Stop\_Time（若存在）为可能的最新日期和时间，使其在测试结束后进行；设置Log\_Enable为FALSE。

测试步骤：

1. WRITE Log\_Enable = TRUE

2. MAKE ( Trend Log object collect number of records specified by Notification\_Threshold)

3. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the Trend Log object being tested),

‘Time Stamp’ = (any appropriate BACnetTimeStamp value),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-NORMAL transition),

‘Event Type’ = BUFFER\_READY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = NORMAL,

‘Event Values’ = (BACnetObjectIdentifier of the IUT’s Device object),

(BACnetObjectIdentifier of the Trend Log object),

(any BACnetDateTime),

(current local BACnetDateTime)

4. TRANSMIT BACnet-SimpleACK-PDU

5. MAKE ( Trend Log object collect number of records specified by Notification\_Threshold)

6. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the Trend Log object being tested),

‘Time Stamp’ = (any appropriate BACnetTimeStamp value),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-NORMAL transition),

‘Event Type’ = BUFFER\_READY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = NORMAL,

‘Event Values’ = (BACnetObjectIdentifier of the IUT’s Device object),

(BACnetObjectIdentifier of the Trend Log object),

(BACnetDateTime sent in step 3),

(current local BACnetDateTime)

7. TRANSMIT BACnet-SimpleACK-PDU

8. WRITE Log\_Enable = FALSE

9. IF ( Previous\_Notify\_Time != Event Value parameter 3 ) THEN

ERROR “Previous\_Notify\_Time value is incorrect.”

10. IF ( Current\_Notify\_Time != Event Value parameter 4 ) THEN

ERROR “Current\_Notify\_Time value is incorrect.”

11. IF ( Event\_Time\_Stamps TO-NORMAL element != Event Value parameter 4 ) THEN

ERROR “Event\_Time\_Stamps value is incorrect.”

1. COV订阅失败测试

所需测试：ReadProperty服务执行测试，10.18; WriteProperty服务执行测试，10.22。

BACnet参考条款：12.25.5,12.25.9和12.25.10。

目的：验证COV订阅失败引起TO-FAULT转换。

测试方案：设置趋势日志为通过来自TD的COV订阅来获取数据。在订阅TD后，暂停趋势日志并检查Event\_State。

配置要求：设置Start\_Time（若存在）为测试开始前的日期和时间；设置Stop\_Time（若存在）为可能的最新日期和时间，使其在测试结束后进行；设置Log\_Enable为FALSE。

测试步骤：

1. VERIFY Event\_State = NORMAL

2. WRITE Log\_Enable = TRUE

3. IF (the IUT uses SubscribeCOV for this Trend Log)

RECEIVE SubscribeCOV-Request,

‘Subscriber Process Identifier’ = (any value),

‘Monitored Object Identifier’ = (any object),

‘Issue Confirmed Notifications’ = (TRUE|FALSE),

‘Lifetime’ = (2 \* COV\_Resubscription\_Interval)

ELSE

RECEIVE SubscribeCOVProperty-Request,

‘Subscriber Process Identifier’ = (any value),

‘Monitored Object Identifier’ = (any object),

‘Issue Confirmed Notifications’ = (TRUE|FALSE),

‘Lifetime’ = (2 \* COV\_Resubscription\_Interval),

‘Monitored Property Identifier’ = (the property to be monitored),

‘COV Increment’ = (Client\_COV\_Increment -- optional)

4. TRANSMIT BACnet-SimpleACK-PDU

5. WAIT COV\_Resubscription\_Interval

6. VERIFY Event\_State = FAULT

1. 日志状态测试

所需测试：ReadProperty服务执行测试，10.21; WriteProperty服务执行测试，10.22。

BACnet参考条款：12.25.14。

目的：验证正确记录了禁用日志和缓冲区清除事件。

测试方案：清除缓冲区。更改Enable属性，验证Record\_Count属性是否已更改，并验证Log\_Buffer是否正确生成状态目录。设置Record\_Count为零，Enable属性为FALSE，验证缓冲区清除的事件是否记录到Log\_Buffer中。

测试配置：设置趋势日志O1为通过任意方式获取数据。设置日志记录，使测试在没有趋势缓冲区溢出的情况下进行。

测试步骤：

1. WRITE Enable = FALSE

2. WRITE Record\_Count = 0

3. VERIFY Record\_Count = 1

4. TRANSMIT ReadRange

‘Object Identifier’ = O1,

‘Property Identifier’ = Log\_Buffer,

‘Reference Index’ = 1,

‘Count’ = 1

5. RECEIVE ReadRange-Ack

‘Object Identifier’ = O1,

‘Property Identifier’ = Log\_Buffer,

‘Result Flags’ = (True, True, False),

‘Item Count’ = 1

‘Item Data’ = ((a buffer purged record))

6. WRITE Enable = TRUE

7. WRITE Enable = FALSE

8. TRANSMIT ReadRange

‘Object Identifier’ = O1,

‘Property Identifier’ = Log\_Buffer,

‘Reference Index’ = 1,

‘Count’ = 2

9. RECEIVE ReadRangeAck

‘Object Identifier’ = O1,

‘Property Identifier’ = Log\_Buffer,

‘Result Flags’ = (True, False, False),

‘Item Count’ = 2

‘Item Data’ = ( (a buffer purged record), (a log-enable record) )

10. TRANSMIT ReadRange

‘Object Identifier’ = O1,

‘Property Identifier’ = Log\_Buffer,

‘Reference Time’ = (2154-12-31, 23:59:59.99),

‘Count’ = -1

11. RECEIVE ReadRangeAck

‘Object Identifier’ = O1,

‘Property Identifier’ = Log\_Buffer,

‘Result Flags’ = (False, True, False),

‘Item Count’ = 1

‘Item Data’ = ( (a log-disable record) )

1. Time\_Change测试

所需测试：ReadRange服务执行测试，10.21;（TimeSynchronization服务执行测试，10.30或UTCTimeSynchronization服务执行测试，10.31）

BACnet参考条款：12.25.14。

目的：验证日志缓冲区正确记录时间更改事件

测试方案：更改设备中的时钟，验证是否有记录指示时钟更改的秒数的记录，或者如果未知是否指示零。如果设备不支持设备对象中的Local\_Time属性或无法更改设备中的时间，则应跳过此测试。

测试配置：设置日志为可通过任何方式获取数据。清除Log\_Buffer，使Record\_Count为0；设置日志记录使测试在没有趋势缓冲区溢出的情况下进行。

测试步骤：

1. WRITE Enable = FALSE

2. WRITE Record\_Count = 0

3. READ currentTime = (Device Object of device that contains the log object), Local\_Time

4. WRITE Enable = TRUE

5. MAKE(the time change on the device by deltaTime where deltaTime >= 1 hour)

6. WRITE Enable = FALSE

7. READ N = Record\_Count

8. REPEAT X = (N down through 1) DO {

TRANSMIT ReadRange

‘Object Identifier’ = O1,

‘Property Identifier’ = Log\_Buffer,

‘Reference Index’ = X,

‘Count’ = 1

RECEIVE ReadRangeAck

‘Object Identifier’ = O1,

‘Property Identifier’ = Log\_Buffer,

‘Result Flags’ = (False, True, False),

‘Item Count’ = 1,

‘Item Data’ = ( (a record. If the record is a time-change record, save the timestamp into TS and the time-change value into TC) )

}

9. CHECK ( TC ~= deltaTime )

10. CHECK ( TS ~= currentTime + deltaTime )

1. COV抽样验证测试

所需测试：ReadRange服务执行测试，10.21

BACnet参考条款：12.25.10-11；13.1。

目的：验证记录样本是基于COV而不是基于间隔。

测试方案：设置趋势日志为基于COV增量的日志。启用日志记录一段时间后检查缓冲区，验证缓冲区中的数据基于COV值而非设置的间隔。

配置要求：设置IUT，使被监测对象COV\_Increment属性值为0.0以外的值；Client\_COV\_Increment设置为0.0或NULL以外的值，或者被监视的属性不是数据类型REAL。

测试步骤：

1. WRITE Enable = FALSE

2. WRITE Record\_Count = 0

3. WRITE Interval = 0

4. WRITE Enable = TRUE

5. WAIT ( 10 seconds )

6. MAKE (monitored property change its value)

7. WAIT ( 60 seconds )

8. MAKE (monitored property change its value)

9. WAIT ( 90 seconds )

10. MAKE (monitored property change its value)

11. WAIT ( 40 seconds )

12. MAKE (monitored property change its value)

13. WAIT **Notification Fail Time**

14. WRITE Enable = FALSE

15. READ N = RecordCount

16. REPEAT X = (1 through 4) {

TRANSMIT ReadRange

‘Object Identifier’ = O1,

‘Property Identifier’ = Log\_Buffer,

‘Reference Index’ = N-5+X,

‘Count’ = 1

RECEIVE ReadRangeAck

‘Object Identifier’ = O1,

‘Property Identifier’ = Log\_Buffer,

‘Result Flags’ = (False, False, False),

‘Item Count’ = 1

‘Item Data’ = ( (one data record storing the timestamp in TS[X]) ) }

17. CHECK( TS[2] - TS[1] ~= 60 seconds )

18. CHECK( TS[3] - TS[2] ~= 90 seconds )

19. CHECK( TS[4] - TS[3] ~= 40 seconds )

1. 外部趋势区间收集测试

目的：验证IUT使用ReadProperty以指定的时间间隔提取外部数据。

测试方案：日志使用轮询从TD获取数据。TD验证ReadProperty请求的接收请求的接收在Log\_Interval设置中。

配置要求：日志在整个测试期间轮询外部趋势。Stop\_When\_Full属性（如果可配置）应设置为FALSE，Enable设置为TRUE。配置TD使其不支持ReadPropertyMultiple的执行。

测试步骤：

1. BEFORE (Log\_Interval)

REPEAT X = (for each property logged) DO

RECEIVE ReadProperty-Request,

‘Object Identifier’ = (object that contains the monitored property)

‘Property Identifier’ = (external property that is being trended)

TRANSMIT ReadProperty-Ack

‘Object Identifier’ = (object that contains the monitored property)

‘Property Identifier’ = (property being monitored)

‘Propert y Value’ = (any value)

2. WAIT (Log Interval)

3. REPEAT X = (for each property logged) DO

RECEIVE ReadProperty-Request,

‘Object Identifier’ = (object that contains the monitored property)

‘Property Identifier’ = (external property that is being trended)

TRANSMIT ReadProperty-Ack

‘Object Identifier’ = (object that contains the monitored property)

‘Property Identifier’ = (property being monitored)

‘Property Value’ = (any value)

4. CHECK (to ensure all properties logged are requested)

[新增第7.3.2.24.x5条，第105页]

7.3.2.24.X5 Last\_Notify\_Record测试

所需测试：ReadProperty服务执行测试，10.15; WriteProperty服务执行测试，10.18。

BACnet参考条款：12.25.19和12.25.26。

目的：验证Last\_Notify\_Record属性是否反映最新报告中发送的值。

测试方案：清除Log\_Buffer。日志收集记录至其发出BUFFER\_READY通知，检查Last\_Notify\_Record属性值。

测试配置：配置日志向TD发送BUFFER\_READY通知。

测试步骤：

1. WRITE Record\_Count = 0

2. READ prev = Last\_Notify\_Record

3. WRITE Enable = TRUE

4. MAKE (Log object collect number of records specified by Notification\_Threshold)

5. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (the configured process identifier)

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (Log object being tested),

‘Time Stamp’ = (any appropriate BACnetTimeStamp value),

‘Notification Class’ = (configured notification class),

‘Priority’ = (value configured to correspond to a TO-NORMAL),

‘Event Type’ = BUFFER\_READY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘FromState’ = NORMAL,

‘To State’ = NORMAL,

‘Event Values’ = ( (the log object), Log\_Buffer,

previous-notification = prev,

current-notification, C1 )

6. TRANSMIT BACnet-SimpleACK-PDU

7. WRITE Enable = FALSE

8. VERIFY Last\_Notify\_Record = C1

1. Records\_Since\_Notification测试

所需测试：ReadProperty服务执行测试，10.15; WriteProperty服务执行测试，10.18。

BACnet参考条款：12.25.18,12.25.19,12.25.22。

目的：验证Records\_Since\_Notification属性反映日志记录的尚未通过BUFFER\_READY通知报告的记录数。

测试方案：清除Log\_Buffer，日志收集记录直至其发出BUFFER\_READY报告，在生成第二个报告之前暂停收集，检查Records\_Since\_Notification属性的值。

测试配置：配置趋势日志向TD发送BUFFER\_READY通知。

测试步骤：

1. WRITE Enable = TRUE

2. WRITE Record\_Count = 0

3. MAKE (Log object collect a sufficient number of records in order to trigger a notification)

4. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (the configured process identifier)

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (Log object being tested),

‘Time Stamp’ = (Any appropriate BACnetTimeStamp value),

‘Notification Class’ = (configured notification class),

‘Priority’ = (value configured to correspond to a TO-NORMAL),

‘Event Type’ = BUFFER\_READY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘FromState’ = NORMAL,

‘To State’ = NORMAL,

‘Event Values’ = ((the log object), Log\_Buffer),

previous-notification,

current-notification, C1)

5. TRANSMIT BACnet-SimpleACK-PDU

6. MAKE (Log object collect N records, such that N < Notification\_Threshold-1)

7. WRITE Enable = FALSE

8. READ T1 = Total\_Record\_Count

9. VERIFY Records\_Since\_Notification = T1 - C1

1. 触发验证测试

所需测试：ReadRange服务执行，10.21;

BACnet参考条款：12.25.27,12.30.12

目的：验证记录的样本是基于触发的Logging\_Type。

测试方案：日志配置为基于TRIGGERED进行记录，启用记录。一段时间后检查缓冲区，验证缓冲区中的数据是否基于触发值。

配置要求：设置IUT使被监测对象的Logging\_Type为TRIGGERED。

测试步骤：

1. WRITE Enable = FALSE

2. WRITE Record\_Count = 0

3. WRITE Enable = TRUE

4. WAIT (10 seconds)

5. WRITE Trigger = TRUE

6. WAIT (20 seconds)

7. WRITE Trigger = TRUE

8. WAIT (40 seconds)

9. WRITE Trigger = TRUE

10. WAIT (30 seconds)

11. WRITE Enable = FALSE

12. READ N = RecordCount

13. REPEAT X = (1 through 4)

TRANSMIT ReadRange

‘Object Identifier’ = O1,

‘Property Identifier’ = Log\_Buffer,

‘Reference Index’ = N-4+X,

‘Count’ = 1

RECEIVE ReadRangeAck

‘Object Identifier’ = O1,

‘Property Identifier’ = Log\_Buffer,

‘Result Flags’ = (False, False, False),

‘Item Count’ = 1

‘Item Data’ = ( (one data record storing the timestamp in TS[X]) )

14. CHECK( TS[2] - TS[1] ~= 20 seconds )

15. CHECK( TS[3] - TS[2] ~= 40 seconds )

16. CHECK( TS[4] - TS[3] ~= 30 seconds )

1. 应用服务启动测试

本节中定义的测试用例被用来验证BACnet设备是否能正确启动指定的应用程序服务。应测试BACnet设备是否能够启动PICS中支持启动的每个应用程序服务。

对本节中包含的每个应用程序服务，定义了几个测试用例，这些例子都测试了在BACnet标准中为服务定义的各种选项和功能。测试用例是在测试执行（IUT）和测试设备（TD）之间交换的一个或多个消息的序列，以便确定特定选项或特征是否被正确实现。将具有相似或相关目的的多个测试用例收集到测试组中。

每个测试用例都介绍了要交换的一个或多个消息的序列。当IUT和TD完全按照测试用例中的描述交换消息时，会传递结果否则测试失败。对本标准中定义的其他测试，只有在执行并被IUT通过时有效。这些所需测试在测试用例描述中注明。

由于本节中测试的目的是测试BACnet服务请求的启动，因此其中许多指示第一步是接收特定服务请求，而非如何使IUT启动预期请求。若供应商提供了一种使IUT发起请求的方法，使IUT采用这些方法的措施是本地问题。

在某些情况下，由于测试功能需要IUT中不支持的特定BACnet对象或可选属性，可能无法证明与特定测试用例的一致性。例如，设备可能支持文件访问服务，但仅限制文件流访问。这样的设备会有无法证明它可以实现文件访问服务的记录访问功能。这种情况下，如果PICS文件清楚地表明了限制，则IUT与BACnet一致；若PICS文件未表明限制，则与BACnet标准不一致。除非由于不受支持的对象或不受支持的可选属性引起冲突，否则BACnet应用程序服务的所有功能和可选参数都应被支持。

* 1. 确认警报（AcknowledgeAlarm）服务启动测试

所需测试：无

BACnet参考条款：14.5

目的：验证IUT是否能够通过ConfirmedEventNotification和UnconfirmedEventNotification服务向IUT报告警报和事件。

配置要求：在TD中选择1个对象O1，该对象配置为向IUT发送事件通知；测试人员设置O1警报状态，以便确认转换。

测试步骤：

1. TRANSMIT ConfirmedEventNotification-Request | UnconfirmedEventNotification-Request,

‘Subscriber Process Identifier’ = (a value acceptable to the IUT configured in the Notification Class object for the IUT),

‘Initiating Device Identifier’ = TD,

‘Event Object Identifier’ = O1,

‘Time Stamp’ = (any valid value, T1),

‘Notification Class’ = (the value configured in O1),

‘Priority’ = (any value selected by the TD),

‘Event Type’ = (any value selected by the TD),

‘Notify Type’ = ALARM | EVENT,

‘AckRequired’ = TRUE,

‘From State’ = (any valid value),

‘To State’ = (any valid value, S1),

‘Event Values’ = (any event values appropriate to the event type)

2. IF (the ConfirmedEventNotification choice was selected) THEN

RECEIVE BACnet-SimpleACK-PDU

3. MAKE (the IUT acknowledge O1)

4. RECEIVE AcknowledgeAlarm-Request,

‘Acknowledging Process Identifier’ = (any process identifier),

‘Event Object Identifier’ = O1,

‘Event State Acknowledged’ = S1,

‘Time Stamp’ = T1,

‘Acknowledgement Source’ = (any valid value),

‘Time of Acknowledgement’ = (any valid value)

5. TRANSMIT BACnet-SimpleACK-PDU

* 1. 确认COV通知（ConfirmedCOVNotification）服务启动测试

本节定义了演示支持启动ConfirmedCOVNotification服务请求所必需的测试。ConfirmedCOVNotification测试针对提供内部COV的报告功能的特定对象类型。 凡支持PICS支持的内部COV报告的每种对象类型，IUT都通过测试。

BACnet参考条款：14.6。

所需测试： SubscribeCOV服务执行测试，10.10; ReadProperty服务执行测试，10.18; WriteProperty服务执行测试，10.22。

1. 模拟输入，模拟输出，和模拟值对象的Present\_Value属性值更改通知

目的：验证IUT是否可以启动ConfirmedCOVNotification服务请求，以发送模拟输入，模拟输出和模拟值对象的Present\_Value属性的更改。

测试方案：订阅COV通知，使用寿命为L，L的设置应小于24小时且可支持完成测试。被监视对象的Present\_Value改变，改变量小于COV增量，验证没有收到COV通知；再改变Present\_Value，改变量大于COV增量，此时接收到通知。可以使用WriteProperty服务或通过其他方式（如更改由模拟输入对象表示的输入信号）来更改Present\_Value。某些情况下可能需要先写入Out\_Of\_Service属性来实现。若无法写入此属性，供应商应提供替代的触发机制来完成此任务。

配置要求：测试前，Out\_Of\_Service属性值为FALSE。

测试步骤：

REPEAT X = (one supported object of each type from the set Analog Input, Analog Output, and Analog Value) DO {

1. TRANSMIT SubscribeCOV-Request,

‘Subscriber Process Identifier’ = (any value ＞ 0 chosen by the TD),

‘Monitored Object Identifier’ = X,

‘Issue Confirmed Notifications’ = TRUE,

‘Lifetime’ = L

2. RECEIVE BACnet-SimpleACK-PDU

3. RECEIVE ConfirmedCOVNotification-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Initiating Device Identifier’ = IUT,

‘Monitored Object Identifier’ = X,

‘Time Remaining’ = (any value appropriate for the Lifetime selected),

‘List of Values’ = (the initial Present\_Value and initial Status\_Flags)

4. TRANSMIT BACnet-SimpleACK-PDU

5. TRANSMIT ReadProperty-Request,

‘Object Identifier’ = X,

‘Property Identifier’ = COV\_Increment

6. RECEIVE BACnet-ComplexACK-PDU,

‘Object Identifier’ = X,

‘Property Identifier’ = COV\_Increment,

‘Property Value’ = (a value "increment" that will be used below)

7. IF (Out\_Of\_Service is writable) THEN

WRITE X, Out\_Of\_Service = TRUE

RECEIVE BACnet-SimpleACK-PDU

BEFORE **Notification Fail Time**

RECEIVE ConfirmedCOVNotification-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Initiating Device Identifier’ = IUT,

‘Monitored Object Identifier’ = X,

‘Time Remaining’ = (any value appropriate for the Lifetime selected),

‘List of Values’ = (the initial Present\_Value and new Status\_Flags)

TRANSMIT BACnet-SimpleACK-PDU

8. IF (Present\_Value is now writable) THEN

WRITE X, Present\_Value = (any value that differs from "initial Present\_Value" by less than "increment")

RECEIVE BACnet-SimpleACK-PDU

ELSE

MAKE (Present\_Value = any value that differs from "initial Present\_Value" by less than "increment")

9. WAIT **NotificationFailTime**

10. CHECK (verify that no COV notification was transmitted)

11. IF (Present\_Value is now writable) THEN

WRITE X, Present\_Value = (any value that differs from "initial Present\_Value" by an amount greater than "increment")

RECEIVE BACnet-SimpleACK-PDU

ELSE

MAKE (Present\_Value = any value that differs from "initial Present\_Value" by an amount greater than "increment")

12. BEFORE NotificationFailTime

RECEIVE ConfirmedCOVNotification-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Initiating Device Identifier’ = IUT,

‘Monitored Object Identifier’ = X,

‘Time Remaining’ = (any value appropriate for the Lifetime selected),

‘List of Values’ = (the new Present\_Value and new Status\_Flags)

13. TRANSMIT BACnet-SimpleACK-PDU

14. TRANSMIT SubscribeCOV-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Monitored Object Identifier’ = X

15. RECEIVE BACnet-SimpleACK-PDU

16. IF (Out\_Of\_Service is writable) THEN

WRITE X, Out\_Of\_Service = FALSE

RECEIVE BACnet-SimpleACK-PDU

1. 模拟输入，模拟输出，和模拟值对象Status\_Flags属性值的更改通知

目的：验证IUT是否可以启动ConfirmedCOVNotification服务请求，以发送模拟输入、模拟输出、模拟值对象的Status\_Flags属性的更改。

测试方案：订阅COV通知，使用寿命为L，L的设置应小于24小时且可支持完成测试。改变被监视对象的Status\_Flags，改变量小于COV增量时验证没有收到COV通知；再改变检测对象的Status\_Flags值，改变量大于COV增量，此时接收到通知。可以使用WriteProperty服务或通过其他方式来更改Status\_Flags值。某些情况下可能需要先写入Out\_Of\_Service属性来实现。若无法写入Out\_Of\_Service属性或Status\_Flags属性，供应商应提供替代的触发机制来完成此任务。

配置要求：测试前，Out\_Of\_Service属性值为FALSE

测试步骤：

REPEAT X = (one supported object of each type from the set Analog Input, Analog Output, and Analog Value) DO {

1. TRANSMIT SubscribeCOV-Request,

‘Subscriber Process Identifier’ = (any value ＞ 0 chosen by the TD),

‘Monitored Object Identifier’ = X,

‘Issue Confirmed Notifications’ = TRUE,

‘Lifetime’ = L

2. RECEIVE BACnet-SimpleACK-PDU

3. RECEIVE ConfirmedCOVNotification-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Initiating Device Identifier’ = IUT,

‘Monitored Object Identifier’ = X,

‘Time Remaining’ = (any value appropriate for the Lifetime selected),

‘List of Values’ = (the initial Present\_Value and initial Status\_Flags)

4. TRANSMIT BACnet-SimpleACK-PDU

5. WRITE X, Out\_Of\_Service = TRUE | WRITE X, Status\_Flags = (a value that differs from "initial Status\_Flags") |

MAKE (Status\_Flags = any value that differs from "initial Status\_Flags")

6. IF (WriteProperty is used in step 5) THEN

RECEIVE BACnet-SimpleACK-PDU

7. BEFORE **NotificationFailTime**

RECEIVE ConfirmedCOVNotification-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Initiating Device Identifier’ = IUT,

‘Monitored Object Identifier’ = X,

‘Time Remaining’ = (any value appropriate for the Lifetime selected),

‘List of Values’ = (the initial Present\_Value and new Status\_Flags)

8. TRANSMIT BACnet-SimpleACK-PDU

9. TRANSMIT SubscribeCOV-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Monitored Object Identifier’ = X

10. RECEIVE BACnet-SimpleACK-PDU

11. IF (Out\_Of\_Service was changed in step 5) THEN

WRITE X, Out\_Of\_Service = FALSE

RECEIVE BACnet-SimpleACK-PDU

1. 二进制输入，二进制输出，和二进制值对象的Present\_Value属性更改通知

目的：验证IUT是否可以启动ConfirmedCOVNotification服务请求，以输送模拟输入、模拟输出、模拟值对象的Status\_Flags属性的更改。

测试方案：订阅COV通知，使用寿命为L，L的设置应小于24小时且可支持完成测试。改变被监视对象的Status\_Flags，改变量小于COV增量时验证没有收到COV通知；再改变Status\_Flags值，改变量大于COV增量，此时接收到通知。可以使用WriteProperty服务或通过其他方式来更改Status\_Flags值。某些情况下可能需要先写入Out\_Of\_Service属性来实现。若无法写入，供应商应提供替代的触发机制来完成此任务。

配置要求：测试时，Out\_Of\_Service属性值为FALSE。

测试步骤：

REPEAT X = (one supported object of each type from the set Binary Input, Binary Output, and Binary Value) DO {

1. TRANSMIT SubscribeCOV-Request,

‘Subscriber Process Identifier’ = (any value ＞ 0 chosen by the TD),

‘Monitored Object Identifier’ = X,

‘Issue Confirmed Notifications’ = TRUE,

‘Lifetime’ = L

2. RECEIVE BACnet-SimpleACK-PDU

3. RECEIVE ConfirmedCOVNotification-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Initiating Device Identifier’ = IUT,

‘Monitored Object Identifier’ = X,

‘Time Remaining’ = (any value appropriate for the Lifetime selected),

‘List of Values’ = (the initial Present\_Value and initial Status\_Flags)

4. TRANSMIT BACnet-SimpleACK-PDU

5. IF (Out\_Of\_Service is writable) THEN

WRITE X, Out\_Of\_Service = TRUE

RECEIVE BACnet-SimpleACK-PDU

BEFORE **Notification Fail Time**

RECEIVE ConfirmedCOVNotification-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Initiating Device Identifier’ = IUT,

‘Monitored Object Identifier’ = X,

‘Time Remaining’ = (any value appropriate for the Lifetime selected),

‘List of Values’ = (the initial Present\_Value and new Status\_Flags)

TRANSMIT BACnet-SimpleACK-PDU

6. IF (Present\_Value is now writable) THEN

WRITE X, Present\_Value = (any value that differs from "initial Present\_Value")

RECEIVE BACnet-SimpleACK-PDU

ELSE

MAKE (Present\_Value = any value that differs from "initial Present\_Value")

7. BEFORE **NotificationFailTime**

RECEIVE ConfirmedCOVNotification-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Initiating Device Identifier’ = IUT,

‘Monitored Object Identifier’ = X,

‘Time Remaining’ = (any value appropriate for the Lifetime selected),

‘List of Values’ = (the new Present\_Value and new Status\_Flags)

8. TRANSMIT BACnet-SimpleACK-PDU

9. TRANSMIT SubscribeCOV-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Monitored Object Identifier’ = X

10. RECEIVE BACnet-SimpleACK-PDU

11. IF (Out\_Of\_Service is writable) THEN

WRITE X, Out\_Of\_Service = FALSE

RECEIVE BACnet-SimpleACK-PDU

1. 二进制输入，二进制输出，和二进制值对象Status\_Flags属性值更改报告

目的：验证IUT是否可以启动ConfirmedCOVNotification服务请求，以输送二进制输入、二进制输出、二进制值对象的Status\_Flags属性的更改。

测试方案：订阅COV通知，使用寿命为L，L的设置应小于24小时且可支持完成测试。改变被监视对象的Status\_Flags属性，改变量小于COV增量，验证没有收到COV通知；再改变Status\_Flags属性，改变量大于COV增量，此时接收到通知。可以使用WriteProperty服务或通过其他方式来更改Status\_Flags值。某些情况下可能需要先写入Out\_Of\_Service属性来实现。若无法写入Out\_Of\_Service属性或Status\_Flags属性，供应商应提供替代的触发机制来完成此任务。

配置要求：测试前，Out\_Of\_Service属性值为FALSE。

测试步骤：

REPEAT X = (one supported object of each type from the set Binary Input, Binary Output, and Binary Value) DO {

1. TRANSMIT SubscribeCOV-Request,

‘Subscriber Process Identifier’ = (any value ＞ 0 chosen by the TD),

‘Monitored Object Identifier’ = X,

‘Issue Confirmed Notifications’ = TRUE,

‘Lifetime’ = L

2. RECEIVE BACnet-SimpleACK-PDU

3. RECEIVE ConfirmedCOVNotification-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Initiating Device Identifier’ = IUT,

‘Monitored Object Identifier’ = X,

‘Time Remaining’ = (any value appropriate for the Lifetime selected),

‘List of Values’ = (the initial Present\_Value and initial Status\_Flags)

4. TRANSMIT BACnet-SimpleACK-PDU

5. WRITE X, Out\_Of\_Service = TRUE | WRITE X, Status\_Flags = (a value that differs from "initial Status\_Flags") |

MAKE (Status\_Flags = any value that differs from "initial Status\_Flags")

6. IF (WriteProperty is used in step 5) THEN

RECEIVE BACnet-SimpleACK-PDU

7. BEFORE **NotificationFailTime**

RECEIVE ConfirmedCOVNotification-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Initiating Device Identifier’ = IUT,

‘Monitored Object Identifier’ = X,

‘Time Remaining’ = (any value appropriate for the Lifetime selected),

‘List of Values’ = (the initial Present\_Value and new Status\_Flags)

8. TRANSMIT BACnet-SimpleACK-PDU

9. TRANSMIT SubscribeCOV-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Monitored Object Identifier’ = X

10. RECEIVE BACnet-SimpleACK-PDU

11. IF (Out\_Of\_Service was changed in step 5) THEN

WRITE X, Out\_Of\_Service = FALSE

RECEIVE BACnet-SimpleACK-PDU

1. 多状态输入、多状态输出、多状态值、生命安全点、或生命安全区对象的Present\_Value属性值更改报告

目的：验证IUT是否可以启动ConfirmedCOVNotification服务请求，以输送多状态输入、多状态输出、多状态值、生命安全点、或生命安全区对象的Present\_Value属性的更改。

测试方案：订阅COV通知，使用寿命为L，L的设置应小于24小时且可支持完成测试。被监视对象的Present\_Value改变，接收到通知。可以使用WriteProperty服务或通过其他方式（如改变对象的输入信号）来更改Status\_Flags值。某些情况下可能需要先写入Out\_Of\_Service属性来实现。若无法写入上述属性，供应商应提供替代的触发机制来完成此任务。

配置要求：测试时，Out\_Of\_Service属性值为FALSE。

测试步骤：

REPEAT X = (one supported object of each type from the set Multi-state Input, Multi-state Output, Multi- state Value, Life Safety Point, and Life Safety Zone) DO {

1. TRANSMIT SubscribeCOV-Request,

‘Subscriber Process Identifier’ = (any value ＞ 0 chosen by the TD),

‘Monitored Object Identifier’ = X,

‘Issue Confirmed Notifications’ = TRUE,

‘Lifetime’ = L

2. RECEIVE BACnet-SimpleACK-PDU

3. RECEIVE ConfirmedCOVNotification-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Initiating Device Identifier’ = IUT,

‘Monitored Object Identifier’ = X,

‘Time Remaining’ = (any value appropriate for the Lifetime selected),

‘List of Values’ = (the initial Present\_Value and initial Status\_Flags)

4. TRANSMIT BACnet-SimpleACK-PDU

5. IF (Out\_Of\_Service is writable) THEN

WRITE X, Out\_Of\_Service = TRUE

RECEIVE BACnet-SimpleACK-PDU

BEFORE **Notification Fail Time**

RECEIVE ConfirmedCOVNotification-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Initiating Device Identifier’ = IUT,

‘Monitored Object Identifier’ = X,

‘Time Remaining’ = (any value appropriate for the Lifetime selected),

‘List of Values’ = (the initial Present\_Value and new Status\_Flags)

TRANSMIT BACnet-SimpleACK-PDU

6. IF (Present\_Value is now writable) THEN

WRITE X, Present\_Value = (any value that differs from "initial Present\_Value")

RECEIVE BACnet-SimpleACK-PDU

ELSE

MAKE (Present\_Value = any value that differs from "initial Present\_Value")

7. BEFORE **NotificationFailTime**

RECEIVE ConfirmedCOVNotification-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Initiating Device Identifier’ = IUT,

‘Monitored Object Identifier’ = X,

‘Time Remaining’ = (any value appropriate for the Lifetime selected),

‘List of Values’ = (the new Present\_Value and new Status\_Flags)

8. TRANSMIT BACnet-SimpleACK-PDU

9. TRANSMIT SubscribeCOV-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Monitored Object Identifier’ = X

10. RECEIVE BACnet-SimpleACK-PDU

11. IF (Out\_Of\_Service is writable) THEN

WRITE X, Out\_Of\_Service = FALSE

RECEIVE BACnet-SimpleACK-PDU

1. 多状态输入、多状态输出、多状态值、生命安全点和生命安全区对象Status\_Flags属性值更改报告

目的：验证IUT是否可以启动ConfirmedCOVNotification服务请求，以输送多状态输入、多状态输出、多状态值、生命安全点、或生命安全区对象的Status\_Flags属性的更改。

测试方案：订阅COV通知，使用寿命为L，L的设置应小于24小时且可支持完成测试。被监视对象的Status\_Flags属性改变，接收到通知。可以使用WriteProperty服务或通过其他方式来更改Status\_Flags值。某些情况下可能需要先写入Out\_Of\_Service属性来实现。若无法写入Out\_Of\_Service属性或Status\_Flags属性，供应商应提供替代的触发机制来完成此任务。

配置要求：测试时，Out\_Of\_Service属性值为FALSE。

测试步骤：

REPEAT X = (one supported object of each type from the set Multi-state input, Multi-state Output, Multi-state Value, Life Safety Point, and Life Safety Zone) DO {

1. TRANSMIT SubscribeCOV-Request,

‘Subscriber Process Identifier’ = (any value ＞ 0 chosen by the TD),

‘Monitored Object Identifier’ = X,

‘Issue Confirmed Notifications’ = TRUE,

‘Lifetime’ = L

2. RECEIVE BACnet-SimpleACK-PDU

3. RECEIVE ConfirmedCOVNotification-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Initiating Device Identifier’ = IUT,

‘Monitored Object Identifier’ = X,

‘Time Remaining’ = (any value appropriate for the Lifetime selected),

‘List of Values’ = (the initial Present\_Value and initial Status\_Flags)

4. TRANSMIT BACnet-SimpleACK-PDU

5. WRITE X, Out\_Of\_Service = TRUE | WRITE X, Status\_Flags = (a value that differs from "initial Status\_Flags") |

MAKE (Status\_Flags = any value that differs from "initial Status\_Flags")

6. IF (WriteProperty is used in step 5) THEN

RECEIVE BACnet-SimpleACK-PDU

7. BEFORE **NotificationFailTime**

RECEIVE ConfirmedCOVNotification-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Initiating Device Identifier’ = IUT,

‘Monitored Object Identifier’ = X,

‘Time Remaining’ = (any value appropriate for the Lifetime selected),

‘List of Values’ = (the initial Present\_Value and new Status\_Flags)

8. TRANSMIT BACnet-SimpleACK-PDU

9. TRANSMIT SubscribeCOV-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Monitored Object Identifier’ = X

10. RECEIVE BACnet-SimpleACK-PDU

11. IF (Out\_Of\_Service was changed in step 5) THEN

WRITE X, Out\_Of\_Service = FALSE

RECEIVE BACnet-SimpleACK-PDU

1. 循环对象Present\_Value属性值更改报告

目的：验证IUT是否可以启动ConfirmedCOVNotification服务请求，以输送循环对象的Present\_Value属性的更改。

测试方案：订阅COV通知，使用寿命为L，L的设置应小于24小时且可支持完成测试。改变被监视对象的Status\_Flags，改变量小于COV增量时验证没有收到COV通知；再改变Status\_Flags值，改变量大于COV增量，此时接收到通知。

可以通过设置Loop Out\_Of\_Service并直接写入Present\_Value来更改Present\_Value属性。若无法写入，供应商应提供替代的触发机制来完成此任务，如更改Setpoint或Setpoint\_Reference。

在下面的测试步骤中，被测试的循环对象的对象标识符为O1。

测试步骤：

1. TRANSMIT SubscribeCOV-Request,

‘Subscriber Process Identifier’ = (any value ＞ 0 chosen by the TD),

‘Monitored Object Identifier’ = LO1,

‘Issue Confirmed Notifications’ = TRUE,

‘Lifetime’ = L

2. RECEIVE BACnet-SimpleACK-PDU

3. RECEIVE ConfirmedCOVNotification-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Initiating Device Identifier’ = IUT,

‘Monitored Object Identifier’ = O1,

‘Time Remaining’ = (any value appropriate for the Lifetime selected),

‘List of Values’ = (the initial Present\_Value, initial Status\_Flags, initial Setpoint, and initial Controlled\_Variable\_Value)

4. TRANSMIT BACnet-SimpleACK-PDU

5. TRANSMIT ReadProperty-Request,

‘Object Identifier’ = O1,

‘Property Identifier’ = COV\_Increment

6. RECEIVE BACnet-ComplexACK-PDU,

‘Object Identifier’ = O1,

‘Property Identifier’ = COV\_Increment,

‘Property Value’ = (a value "increment" that will be used below)

7. IF (Out\_Of\_Service is writable) THEN

WRITE O1, Out\_Of\_Service = TRUE

BEFORE **Notification Fail Time**

RECEIVE ConfirmedCOVNotification-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Initiating Device Identifier’ = IUT,

‘Monitored Object Identifier’ = O1,

‘Time Remaining’ = (any value appropriate for the Lifetime selected),

‘List of Values’ = (the initial Present\_Value, new Status\_Flags, initial

Setpoint, and initial Controlled\_Variable\_Value)

TRANSMIT BACnet-SimpleACK-PDU

8. IF (Present\_Value is now writable) THEN

WRITE O1, Present\_Value = (any value that differs from "initial Present\_Value" by less than "increment")

ELSE

MAKE (Present\_Value = any value that differs from "initial Present\_Value" by less than "increment")

9. WAIT **NotificationFailTime**

10. CHECK (verify that no COV notification was transmitted)

11. IF (Present\_Value is now writable) THEN

WRITE O1, Present\_Value = (any value that differs from "initial Present\_Value" by an amount greater than "increment")

ELSE

MAKE (Present\_Value = any value that differs from "initial Present\_Value" by an amount greater than "increment")

12. BEFORE **NotificationFailTime**

RECEIVE ConfirmedCOVNotification-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Initiating Device Identifier’ = IUT,

‘Monitored Object Identifier’ = O1,

‘Time Remaining’ = (any value appropriate for the Lifetime selected),

‘List of Values’ = (the new Present\_Value, new Status\_Flags, initial

Setpoint, and initial Controlled\_Variable\_Value)

13. TRANSMIT BACnet-SimpleACK-PDU

14. TRANSMIT SubscribeCOV-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Monitored Object Identifier’ = O1

15. RECEIVE BACnet-SimpleACK-PDU

16. IF (Out\_Of\_Service is writable) THEN

WRITE L, Out\_Of\_Service = FALSE

1. 循环对象Status\_Flags属性值更改报告

目的：验证IUT可以启动ConfirmedCOVNotification服务请求，以输送循环对象Status\_Flags属性的更改。

测试方案：订阅COV通知，使用寿命为L，L的设置应小于24小时且可支持完成测试。改变被监视对象的Status\_Flags属性，接收到通知。可以通过写入Present\_Value或其他方式来更改循环对象Status\_Flags属性。在某些情况下，写入Out\_Of\_Service属性即可完成此测试。若无法写入Status\_Flags或Out\_Of\_Service，供应商应提供替代的触发机制来完成此任务。

在下面的测试步骤中，被测试的循环对象的对象标识符为O1。

配置要求：测试时，Out\_Of\_Service属性值为FALSE。

测试步骤：

1. TRANSMIT SubscribeCOV-Request,

‘Subscriber Process Identifier’ = (any value ＞ 0 chosen by the TD),

‘Monitored Object Identifier’ = O1,

‘Issue Confirmed Notifications’ = TRUE,

‘Lifetime’ = L

2. RECEIVE BACnet-SimpleACK-PDU

3. RECEIVE ConfirmedCOVNotification-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Initiating Device Identifier’ = IUT,

‘Monitored Object Identifier’ = O1,

‘Time Remaining’ = (any value appropriate for the Lifetime selected),

‘List of Values’ = (the initial Present\_Value, initial Status\_Flags, initial Setpoint, and initial Controlled\_Variable\_Value)

4. TRANSMIT BACnet-SimpleACK-PDU

5. WRITE O1, Out\_Of\_Service = TRUE | WRITE O1, Status\_Flags = (a value that differs from "initial Status\_Flags") |

MAKE (Status\_Flags = any value that differs from "initial Status\_Flags")

6. IF (WriteProperty is used in step 5) THEN

RECEIVE BACnet-SimpleACK-PDU

7. BEFORE **NotificationFailTime**

RECEIVE ConfirmedCOVNotification-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Initiating Device Identifier’ = IUT,

‘Monitored Object Identifier’ = O1,

‘Time Remaining’ = (any value appropriate for the Lifetime selected),

‘List of Values’ = (the initial Present\_Value, new Status\_Flags, initial Setpoint, and initial Controlled\_Variable\_Value)

8. TRANSMIT BACnet-SimpleACK-PDU

9. TRANSMIT SubscribeCOV-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Monitored Object Identifier’ = O1

10. RECEIVE BACnet-SimpleACK-PDU

11. IF (Out\_Of\_Service was changed in step 5) THEN

WRITE L, Out\_Of\_Service = FALSE

RECEIVE BACnet-SimpleACK-PDU

1. 缺失Lifetime测试

目的：此测试用例验证SubscribeCOV的一种特殊情况——缺少Lifetime参数即无限订阅Lifetime。

测试方案：IUT订阅COV报告，订阅消息省略Lifetime参数。从IUT接收COV报告，验证Time Remaining值为0。

测试步骤：

1. TRANSMIT SubscribeCOV-Request,

‘Subscriber Process Identifier’ = (any value > 0 chosen by the TD),

‘Monitored Object Identifier’ = X,

‘Issue Confirmed Notifications’ = TRUE

2. RECEIVE BACnet-SimpleACK-PDU,

3. RECEIVE ConfirmedCOVNotification-Request,

‘Subscriber Process Identifier’ = (the same value used in step 1),

‘Initiating Device Identifier’ = IUT,

‘Monitored Object Identifier’ = X,

‘Time Remaining’ = 0,

‘List of Values’ = (the initial Present\_Value and initial Status\_Flags)

4. TRANSMIT BACnet-SimpleACK-PDU

* 1. 未确认COV通知（UnconfirmedCOVNotification）服务启动测试

本节规定义了演示支持启动UnconfirmedCOVNotification服务测试请求所必需的测试。UnconfirmedCOVNotification服务启动测试针对提供内在COV的报告功能特定的对象类型。凡支持PICS支持的内部COV报告的对象类型，IUT都通过测试。

所需测试： SubscribeCOV服务执行测试，10.10; ReadProperty服务执行测试，10.18;WriteProperty服务执行测试，10.22。

BACnet参考条款：14.7。

1. 模拟输入，模拟输出，和模拟值对象的Present\_Value 属性值更改通知

目的：验证IUT是否可以启动UnconfirmedCOVNotification服务请求，以输送模拟输入，模拟输出和模拟值对象的Present\_Value属性的更改。

测试步骤：此测试用例的步骤与9.2.1中的测试步骤大致相同，将步骤1中的SubscribeCOV服务请求中Issue Confirmed Notifications参数的值设为FALSE，所有ConfirmedCOVNotification请求改为UnconfirmedCOVNotification请求，且不承认未经证实的服务。使用MAC地址报告消息，TD为接收者。

1. 模拟输入，模拟输出，和模拟值对象的Status\_Flags属性值更改通知

目的：验证IUT是否可以启动UnconfirmedCOVNotification服务请求，以输送模拟输入、模拟输出、模拟值对象的Status\_Flags属性的更改。

测试步骤：此测试用例的步骤与9.2.2中的测试步骤大致相同，将步骤1中的SubscribeCOV服务请求中Issue Confirmed Notifications参数的值设为FALSE，所有ConfirmedCOVNotification请求改为UnconfirmedCOVNotification请求，且不承认未经证实的服务。使用MAC地址报告消息，TD成为接收者。

1. 二进制输入，二进制输出，和二进制值对象的Present\_Value属性值更改通知

目的：验证IUT是否可以启动UnconfirmedCOVNotification服务请求，以输送二进制输入、二进制输出、二进制值对象的Present\_Value属性更改。

测试步骤：此测试用例的步骤与9.2.3中的测试步骤大致相同，将步骤1中的SubscribeCOV服务请求中Issue Confirmed Notifications参数的值设为FALSE，所有ConfirmedCOVNotification请求改为UnconfirmedCOVNotification请求，且不承认未经证实的服务。使用MAC地址报告消息，TD为接收者。

1. 二进制输入，二进制输出，和二进制值对象的Status\_Flags属性值更改通知

目的：验证IUT是否可以启动UnconfirmedCOVNotification服务请求，以输送二进制输入、二进制输出、二进制值对象的Status\_Flags属性的更改。

测试步骤：此测试用例的步骤与9.2.4中的测试步骤大致相同，将步骤1中的SubscribeCOV服务请求中Issue Confirmed Notifications参数的值设为FALSE，所有ConfirmedCOVNotification请求改为UnconfirmedCOVNotification请求，且不承认未经证实的服务。使用MAC地址报告消息，TD为接收者。

1. 多状态输入、多状态输出、多状态值、生命安全点和生命安全区对象的Present\_Value属性值更改通知

目的：验证IUT是否可以启动UnconfirmedCOVNotification服务请求，以发送多状态输入、多状态输出、多状态值、生命安全点和生命安全区对象的Present\_Value属性的更改。

测试步骤：此测试用例的步骤与9.2.5中的测试步骤大致相同，将步骤1中的SubscribeCOV服务请求中Issue Confirmed Notifications参数的值设为FALSE，所有ConfirmedCOVNotification请求改为UnconfirmedCOVNotification请求，且不承认未经证实的服务。使用MAC地址报告消息，TD为接收者。

1. 多状态输入、多状态输出、多状态值、生命安全点和生命安全区对象的Status\_Flags属性值更改通知

目的：验证IUT是否可以启动UnconfirmedCOVNotification服务请求，以发送多状态输入、多状态输出、多状态值、生命安全点和生命安全区对象的Status\_Flags属性的更改。

测试步骤：此测试用例的步骤与9.2.6中的测试步骤大致相同，将步骤1中的SubscribeCOV服务请求中Issue Confirmed Notifications参数的值设为FALSE，所有ConfirmedCOVNotification请求改为UnconfirmedCOVNotification请求，且不承认未经证实的服务。使用MAC地址报告消息，TD为接收者。

1. 循环对象的Present\_Value属性值更改通知

目的：验证IUT是否可以启动UnconfirmedCOVNotification服务请求，以发送循环对象的Present\_Value属性的更改。

测试步骤：此测试用例的步骤与9.2.7中的测试步骤大致相同，将步骤1中的SubscribeCOV服务请求中Issue Confirmed Notifications参数的值设为FALSE，所有ConfirmedCOVNotification请求改为UnconfirmedCOVNotification请求，且不承认未经证实的服务。使用MAC地址报告消息，TD为接收者。

1. 循环对象的Status\_Flags属性值更改通知

目的：验证IUT是否可以启动UnconfirmedCOVNotification服务请求，以发送循环对象的Status\_Flags属性的更改。

测试步骤：此测试用例的步骤与9.2.8中的测试步骤大致相同，将步骤1中的 SubscribeCOV服务请求中Issue Confirmed Notifications参数的值设为FALSE，所有ConfirmedCOVNotification请求改为UnconfirmedCOVNotification请求，且不承认未经证实的服务。使用MAC地址报告消息，TD做接收者。

1. 退订的值更改通知

退订COV报告与使用UnconfirmedCOVNotification服务的订阅COV报告有两点不同：不需订阅；Subscriber Process Identifier参数值通常为零。

BACnet参考条款：13.7。

目的：验证IUT是否可以在未订阅COV通知时启动UnconfirmedCOVNotification服务请求。

测试方案：配置IUT，发送退订COV报告，TD等待报告。由于没有定义触发器，若报告不能定期发送，供应商应告知测试人员如何生成报告。

测试步骤：

1. MAKE (the IUT send an unsubscribed COV notification)

2. RECEIVE UnconfirmedCOVNotification-Request,

‘Subscriber Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Monitored Object Identifier’ = (any valid object identifier),

‘Time Remaining’ = 0,

‘List of Values’ = (any valid properties and values from the monitored object)

1. 设备重启报告

目的：验证IUT在重置时对其Restart\_Notification\_Recipients属性中的每个条目启动UnconfirmedCOVNotification服务请求。

测试方案：配置IUT发送重启报告，然后重置IUT，TD检查重启报告。

设备重启报告与使用UnconfirmedCOVNotification服务的订阅COV通知有两点不同：通过Restart\_Notification\_Recipients属性而非SubscribeCOV进行订阅；Subscriber Process Identifier参数值通常为零。

配置要求：对于设备形式的IUT每个接受者的Restart\_Notification\_Recipients property属性，网络上都应该有一个设备可以回答Who-Is请求，以便IUT在发送重新启动通知之前可以确定寻址信息。

测试步骤：

1. IF (Restart\_Notification\_Recipients is writable) THEN

WRITE(Restart\_Notification\_Recipients = any non-empty list of Recipients)

ELSE

MAKE (Restart\_Notification\_Recipients contain any non-empty list of Recipients)

2. READ T1 = Local\_Time

3. MAKE(the IUT reset)

4. REPEAT X = (each entry in the Restart\_Notification\_Recipients) DO {

BEFORE **Notification Fail Time**

RECEIVE UnconfirmedCOVNotification-Request,

DESTINATION = X,

‘Subscriber Process Identifier’ = 0,

‘Initiating Device Identifier’ = IUT,

‘Monitored Object Identifier’ = (the IUT Device Identifier),

‘Time Remaining’ = 0,

‘List of Values’ = (System\_Status=OPERATIONAL,

Time\_Of\_Device\_Restart = (T2),

Last\_Restart\_Reason=(any valid restart reason, R))

}

5. VERIFY Time\_Of\_Device\_Restart = T2

6. CHECK (T1 ~= T2)

7. VERIFY Last\_Restart\_Reason = R

注意事项：并非所有IUT都可以准确区分重启的原因的类型，因此步骤4中返回的值没有要求。无论IUT在步骤4中生成UnconfirmedCOVNotification请求顺序如何，都通过测试。步骤4中发出所有报告的T2值应相同。

* 1. 确认事件通知（ConfirmedEventNotification）服务启动测试

本节规定了执行支持ConfirmedEventNotification服务测试请求所必需的测试。ConfirmedEventNotification测试特定于所使用的事件检测算法。对于支持内在事件报告的每种对象类型，IUT必须通过适用于该对象类型的事件检测算法的测试。如果IUT支持事件注册Event Enrollment对象，则它必须通过与每种支持的事件类型相对应的事件检测算法的测试。

1. 位串更改测试

所需测试：ReadProperty服务执行测试，10.18; WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.5,12.12.7,13.3.1和13.8。

目的：验证位串事件算法更改的正确操作。此测试适用于Event\_Type为CHANGE\_OF\_BITSTRING的Event Enrollment对象。

测试方案：对象初始状态为NORMAL。参考属性更改为List\_Of\_Bitstring\_Values中指定的值之一。 时间延迟到期，对象进入OFFNORMAL状态并发送事件通知消息，参考属性变为与NORMAL状态对应的值。在时间延迟后，对象进入NORMAL状态并发送事件通知消息。

配置要求：配置IUT,使Event\_Enable值为TRUE（用于TO-OFFNORMAL、TO-NORMAL转换）；Issue\_Confirmed\_Notifications属性值为TRUE；事件生成对象初始态为NORMAL。

测试步骤：

1. VERIFY Event\_State = NORMAL

2. IF (the referenced property is writable) THEN

WRITE (referenced property) = (a value x: x = one of the List\_Of\_Bitstring\_Values after the bitmask is applied)

ELSE

MAKE (the referenced property have a value x: x = one of the List\_Of\_Bitstring\_Values after the bitmask is applied)

3. WAIT (Time\_Delay)

4. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO- OFFNORMAL transition),

‘Event Type’ = CHANGE\_OF\_BITSTRING,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = OFFNORMAL,

‘ Event Values’ = referenced-bitstring, Status\_Flags

5. TRANSMIT BACnet-SimpleACK-PDU

6. VERIFY STATUS\_FLAGS = (TRUE, FALSE, ?, ?)

7. VERIFY Event\_State = OFFNORMAL

8. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

VERIFY Event\_Time\_Stamps = (the timestamp in step 4, \*, \*)

9. IF (Present\_Value is writable) THEN

WRITE (referenced property) = (a value x: x corresponds to a NORMAL state)

ELSE

MAKE (the referenced property have a value x: x corresponds to a NORMAL state)

10. WAIT (Time\_Delay)

11. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-NORMAL transition),

‘Event Type’ = CHANGE\_OF\_BITSTRING,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = OFFNORMAL,

‘To State’ = NORMAL,

‘ Event Values’ = referenced-bitstring, Status\_Flags

12. TRANSMIT BACnet-SimpleACK-PDU

13. VERIFY STATUS\_FLAGS = (FALSE, FALSE, ?, ?)

14. VERIFY Event\_State = NORMAL

15. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

VERIFY Event\_Time\_Stamps = (the timestamp in step 4, \*, the timestamp in step 11)

注意事项：测试描述中省略了Message Text参数，因为它是可选的。IUT报告消息中可能包括该参数。 在步骤8和15中由“\*”指示的时间戳的值可以表示在步骤4中的时间戳之前的时间值或表示未指定时间。

1. 状态变化测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.6,12.8,12.18,12.20,13.2,13.3.2和13.8。

目的：验证CHANGE\_OF\_STATE事件算法的正确操作。此测试适用于Event\_Type为CHANGE\_OF\_STATE的Event Enrollment对象，以及二进制输入、二进制值、多状态输入和多状态值对象的内部事件报告。

测试方案：对象初始态为NORMAL。Present\_Value（参考属性）更改为List\_Of\_Values中指定的值之一。在时间延迟到期时，对象应进入OFFNORMAL状态并发送事件通知消息，Present\_Value（参考属性）改变为对应于NORMAL状态的值。在时间延迟之后，对象应进入NORMAL状态并发送事件通知消息。对于多状态输入和多状态值对象，有一种特殊的情况，即CHANGE\_OF\_STATE算法适用于转换到FAULT状态。以下测试程序包括此特殊情况的测试。

配置要求：配置IUT,使Event\_Enable属性值为TRUE（用于TO-OFFNORMAL、TO-FAULT和TO-NORMAL转换）；Issue\_Confirmed\_Notifications属性值为TRUE；事件生成对象初始态为NORMAL。

在下述测试中，Present\_Value用作参考属性。若正在测试事件注册对象，则应使用适当的参考属性替换Present\_Value属性。

测试步骤：

1. VERIFY Event\_State = NORMAL

2. IF (the object, or referenced object, if using Event Enrollment, is a binary object or it is a multi-state object with a non-empty Alarm\_Values property) THEN

3. IF (Present\_Value is writable) THEN

WRITE Present\_Value = (a value x: x = Alarm\_Value for binary objects or one of the Alarm\_Values for multi-state objects)

ELSE

MAKE (Present\_Value have a value x: x = Alarm\_Value for binary objects or one of the Alarm\_Values for multi-state objects)

4. WAIT (Time\_Delay)

5. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollmentobject being tested),

‘Time Stamp’ = (T1, the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO- OFFNORMAL transition),

‘Event Type’ = CHANGE\_OF\_STATE,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = OFFNORMAL,

‘ Event Values’ = Present\_Value, Status\_Flags

6. TRANSMIT BACnet-SimpleACK-PDU

7. IF (the object being tested is NOT an Event Enrollment object) THEN

VERIFY STATUS\_FLAGS = (TRUE, FALSE, ?, ?)

8. VERIFY Event\_State = OFFNORMAL

9. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

VERIFY Event\_Time\_Stamps = (T1, \*, \*)

10. IF (Present\_Value is writable) THEN

WRITE Present\_Value = (a value x: x corresponds to a NORMAL state)

ELSE

MAKE (Present\_Value have a value x: x corresponds to a NORMAL state)

11. WAIT (Time\_Delay)

12. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (T2, the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO- NORMAL transition),

‘Event Type’ = CHANGE\_OF\_STATE,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = OFFNORMAL,

‘To State’ = NORMAL,

‘ Event Values’ = Present\_Value, Status\_Flags

13. TRANSMIT BACnet-SimpleACK-PDU

14. IF (the object being tested is NOT an Event Enrollment object) THEN

VERIFY STATUS\_FLAGS = (FALSE, FALSE, ?, ?)

15. VERIFY Event\_State = NORMAL

16. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

VERIFY Event\_Time\_Stamps = (T1, \*, T2)

17. IF (the object being tested is a multi-state object that supports intrinsic reporting) THEN

18. IF (Present\_Value is writable) THEN

WRITE Present\_Value = (a value x: x = one of the Fault\_Values)

ELSE

MAKE (Present\_Value have a value x: x = one of the Fault\_Values)

19. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested),

‘Time Stamp’ = (T3, the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO- FAULT transition),

‘Event Type’ = CHANGE\_OF\_STATE,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = FAULT,

‘ Event Values’ = Present\_Value, Status\_Flags

20. TRANSMIT BACnet-SimpleACK-PDU

21. IF (the object being tested is NOT an Event Enrollment object) THEN

VERIFY STATUS\_FLAGS = (FALSE, TRUE, ?, ?)

22. VERIFY Event\_State = FAULT

23. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

VERIFY Event\_Time\_Stamps = (T1, T3, T2)

24. IF (the object being tested is a multi-state object that supports intrinsic reporting and Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

VERIFY Reliability = MULTI\_STATE\_FAULT

25. IF (Present\_Value is writable) THEN

WRITE Present\_Value = (a value x: x corresponds to a NORMAL state)

ELSE

MAKE (Present\_Value have a value x: x corresponds to a NORMAL state)

26. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested),

‘Time Stamp’ = (T4, the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO- NORMAL transition),

‘Event Type’ = CHANGE\_OF\_STATE,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = FAULT,

‘To State’ = NORMAL,

‘ Event Values’ = Present\_Value, Status\_Flags

27. TRANSMIT BACnet-SimpleACK-PDU

28. IF (the object being tested is NOT an Event Enrollment object) THEN

VERIFY STATUS\_FLAGS = (FALSE, FALSE, ?, ?)

29. VERIFY Event\_State = NORMAL

30. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

VERIFY Event\_Time\_Stamps = (T1, T3, T4)

注意事项：测试描述中省略了Message Text参数，因为它是可选的。 IUT报告消息中可能包括该参数。由“\*”指示的时间戳的值可以表示未指定时间或者在时间戳T1之前的时间。

1. 改变值测试

本节定义了演示支持CHANGE\_OF\_VALUE事件算法的所必需的测试。CHANGE\_OF\_VALUE算法可以应用于数值和位串数据类型。IUT应对这两种情况都进行测试。

1. 数值算法

本节测试适用于使用整数或实数数据类型的CHANGE\_OF\_VALUE算法。

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12,13.3.2和13.8。

目的：验证应用于几种数据类型的CHANGE\_OF\_VALUE事件算法的正确操作。此测试适用于Event\_Type为CHANGE\_OF\_VALUE的Event Enrollment对象。

测试方案：对象初始态为NORMAL。参考属性值改为小于Referenced\_Property\_Increment的值，测试人员验证没有传输事件通知；参考属性再更改为与原始值不同且大于Referenced\_Property\_Increment.的值，测试人员验证是否发送了事件通知消息及Event\_State转换是否正确。

配置要求：配置IUT，使Event\_Enable属性值为TRUE（用于TO\_NORMAL转换）；Issue\_Confirmed\_Notifications属性值为TRUE；事件生成对象初始态为NORMAL。

测试步骤：

1. VERIFY Event\_State = NORMAL

2. IF (the referenced property is writable) THEN

WRITE (referenced property) = (a value x: x differs from the initial value by less than Referenced\_Property\_Increment)

ELSE

MAKE (the referenced property have a value x: x differs from the initial value by less than Referenced\_Property\_Increment)

3. WAIT (Time\_Delay + **Notification Fail Time**)

4. CHECK (verify that no event notification message is transmitted)

5. IF (the referenced property is writable) THEN

WRITE (referenced property) = (a value x: x differs from the initial value in step 1 by more than Referenced\_Property\_Increment)

ELSE

MAKE (the referenced property have a value x: x differs from the initial value in step 1 by more than Referenced\_Property\_Increment)

6. WAIT (Time\_Delay)

7. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-NORMAL transition),

‘Event Type’ = CHANGE\_OF\_VALUE,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = NORMAL,

‘ Event Values’ = changed-value, Status\_Flags

8. TRANSMIT BACnet-SimpleACK-PDU

9. VERIFY STATUS\_FLAGS = (FALSE, FALSE, ?, ?)

10. VERIFY Event\_State = NORMAL

11. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

VERIFY Event\_Time\_Stamps = (\*, \*, the timestamp in step 7)

注意事项：测试描述中省略了Message Text参数，因为它是可选的。IUT报告消息中可能包括该参数。在步骤11中由“\*”指示的时间戳可表示在步骤7中的时间戳之前的时间值或未指定时间值。

1. 位串算法

本节中的测试适用于应用于位串数据类型的CHANGE\_OF\_VALUE算法。

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12,13.3.2和13.8。

目的：验证应用于位串数据类型的CHANGE\_OF\_VALUE事件算法的正确操作。此测试适用于Event\_Type为CHANGE\_OF\_VALUE的Event Enrollment对象。

测试方案：对象初始态为NORMAL。参考属性将更改为新值，以使“位掩码”中的所有位均未更改。测试人员验证没有传输事件通知；再次将参考属性更改为一个值，该值与位掩码中包含的一个或多个位不同，测试人员验证是否发送了事件通知消息且Event\_State转换是否正确。

配置要求：配置IUT，使Event\_Enable属性值为TRUE（用于TO\_NORMAL转换）；Issue\_Confirmed\_Notifications属性值为TRUE。配置位掩码，使掩码中包含参考属性的至少一个但不是全部的位。事件生成对象初始态为NORMAL。

测试步骤：

1. VERIFY Event\_State = NORMAL

2. IF (the referenced property is writable) THEN

WRITE (referenced property) = (a value x: x differs from the initial value but only in bits that are not included in Bitmask)

ELSE

MAKE (the referenced property have a value x: x differs from the initial value but only in bits that are not included in Bitmask)

3. WAIT (Time\_Delay + **Notification Fail Time**)

4. CHECK (verify that no event notification message is transmitted)

5. IF (the referenced property is writable) THEN

WRITE (referenced property) = (a value x: x differs from the initial value in one or more bits included in Bitmask)

ELSE

MAKE (the referenced property have a value x: x differs from the initial value one or more bits included in Bitmask)

6. WAIT (Time\_Delay)

7. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-NORMAL transition),

‘Event Type’ = CHANGE\_OF\_VALUE,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = NORMAL,

‘ Event Values’ = changed-value, Status\_Flags

8. TRANSMIT BACnet-SimpleACK-PDU 9. VERIFY STATUS\_FLAGS = (FALSE, FALSE, ?, ?)

10. VERIFY Event\_State = NORMAL

11. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

VERIFY Event\_Time\_Stamps = (\*, \*, the timestamp in step 7)

注意事项：测试描述中省略了Message Text参数，因为它是可选的。IUT报告消息中可能包括该参数。在步骤11中由“\*”指示的时间戳可表示在步骤7中的时间戳之前的时间值或未指定时间值。

1. 命令失败测试

所需测试：ReadProperty服务执行测试，10.18; WriteProperty服务执行测试，10.22。

BACnet参考条款：12.7,12.12,12.19,13.2,13.3.4和13.8。

目的：验证COMMAND\_FAILURE算法的操作。此测试适用于Event\_Type为COMMAND\_FAILURE的Event Enrollment对象以及二进制输出和多状态输出对象的内部事件报告。

测试方案：Feedback\_Value（Feedback\_Property\_Reference）应与常用于验证输出的输入信号去耦。开始时Present\_Value（参考属性）和 Feedback\_Value（Feedback\_Property\_Reference）一致，改变Present\_Value（参考属性）并发送事件报告，指示转换到OFFNORMAL状态。再将 Feedback\_Value（Feedback\_Property\_Reference）更改与Present\_Value（参考属性）一致，发送第二事件通知，指示返回到NORMAL状态。

配置要求：配置IUT，使Event\_Enable属性值为TRUE（用于TO-OFFNORMAL和TO-NORMAL转换）；Issue\_Confirmed\_Notifications属性值为TRUE；事件生成对象初始态为NORMAL；Feedback\_Value属性与用于验证输出的输入信号解耦以便于独立操作。

在下述测试中，Present\_Value用作参考属性，而 Feedback\_Value用于验证输出。若测试Event Enrollment对象，则这些属性应替换为相应的参考属性。

测试步骤：

1. VERIFY Event\_State = NORMAL

2. VERIFY STATUS\_FLAGS = (FALSE, FALSE, FALSE, FALSE)

3. IF (Present\_Value is writable) THEN

WRITE Present\_Value = (a different value)

ELSE

MAKE (Present\_Value take on a different value)

4. WAIT (Time\_Delay)

5. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-OFFNORMAL transition),

‘Event Type’ = COMMAND\_FAILURE,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = OFFNORMAL,

‘ Event Values’ = Present\_Value, Status\_Flags, Feedback\_Value

6. TRANSMIT BACnet-SimpleACK-PDU

7. VERIFY STATUS\_FLAGS = (TRUE, FALSE, ?, ?)

8. VERIFY Event\_State = OFFNORMAL

9. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

VERIFY Event\_Time\_Stamps = (the timestamp in step 5, \*, \*)

10. IF (Feedback\_Value is writable) THEN

WRITE Feedback\_Value = (a value consistent with Present\_Value)

ELSE

MAKE (Feedback\_Value take on a value consistent with Present\_Value)

11. WAIT (Time\_Delay)

12. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-NORMAL transition),

‘Event Type’ = COMMAND\_FAILURE,

‘Notify Type’ = EVENT | ALARM, ‘AckRequired’ = TRUE | FALSE,

‘From State’ = OFFNORMAL,

‘To State’ = NORMAL,

‘ Event Values’ = Present\_Value, Status\_Flags, Feedback\_Value

13. TRANSMIT BACnet-SimpleACK-PDU

14. VERIFY STATUS\_FLAGS = (FALSE, FALSE, ?, ?)

15. VERIFY Event\_State = NORMAL

16. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

VERIFY Event\_Time\_Stamps = (the timestamp in step 5, \*, the timestamp in step 12)

注意事项：测试描述中省略了Message Text参数，因为它是可选的。 IUT报告消息中可能包括该参数。在步骤9和16中由“\*”指示的时间戳可表示在步骤5的时间戳之前的时间值或未指定时间值。

1. FLOATING\_LIMIT浮动限度测试

所需测试：ReadProperty服务执行测试，10.18;WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12,12.17,12.21,13.2,13.3.5和13.8。

目的：验证浮动限制事件算法。此测试适用于Event\_Type为FLOATING\_LIMIT的Event Enrollment对象以及支持内部报告的循环对象。测试循环对象时，High\_Diff\_Limit和Low\_Diff\_Limit都将被下述测试的Error\_Limit替换。

测试方案：对象初始态为NORMAL。参考属性值设置为较低但处于死区上限内的值，此时对象仍应处于NORMAL状态。将参考属性设置为高于上限的值，在时间延迟到期后，对象进入HIGH\_LIMIT状态并发送事件通知消息。将参考属性降低到低于上限但仍在限制的死区范围内的值，对象保持HIGH\_LIMIT状态。将参考属性进一步设置到不在限制死区范围内的正常值，延迟到期后，对象进入NORMAL状态并发出事件通知。重复相同的过程以测试出下限。

配置要求：配置IUT，使Event\_Enable属性值为TRUE（用于TO-OFFNORMAL和TO-NORMAL转换）；Issue\_Confirmed\_Notifications属性值为TRUE；事件生成对象初始态为NORMAL。

1. VERIFY Event\_State = NORMAL

2. IF (the referenced property is writable) THEN

WRITE (referenced property) = (a value x:

(Setpoint\_Reference + High\_Diff\_Limit – Deadband)< x <(Setpoint\_Reference + High\_Diff\_Limit))

ELSE

MAKE (the referenced property have a value x:

(Setpoint\_Reference + High\_Diff\_Limit – Deadband)< x <(Setpoint\_Reference + High\_Diff\_Limit))

3. WAIT (Time\_Delay + **Notification Fail Time**)

4. CHECK (verify that no notification message has been transmitted)

5. VERIFY Event\_State = NORMAL

6. IF (the referenced property is writable) THEN

WRITE (referenced property) = (a value x: x > (Setpoint\_Reference + High\_Diff\_Limit))

ELSE

MAKE (the referenced property have a value x: x > (Setpoint\_Reference + High\_Diff\_Limit))

7. WAIT (Time\_Delay)

8. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the Loop object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-OFFNORMAL transition),

‘Event Type’ = FLOATING\_LIMIT,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = HIGH\_LIMIT,

‘ Event Values’ = reference-value, Status\_Flags, setpoint-value, error-limit,

9. TRANSMIT BACnet-SimpleACK-PDU

10. VERIFY STATUS\_FLAGS = (TRUE, FALSE, ?, ?)

11. VERIFY Event\_State = HIGH\_LIMIT

12. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

VERIFY Event\_Time\_Stamps = (the timestamp in step 8, \*, \*)

13. IF (the referenced property is writable) THEN

WRITE (referenced property) = (a value x:

(Setpoint\_Reference + High\_Diff\_Limit – Deadband)< x < Setpoint\_Reference + High\_Diff\_Limit))

ELSE

MAKE (the referenced property have a value x:

(Setpoint\_Reference + High\_Diff\_Limit – Deadband)< x < Setpoint\_Reference + High\_Diff\_Limit))

14. WAIT (Time\_Delay + **Notification Fail Time**)

15. CHECK (verify that no notification message has been transmitted)

16. VERIFY Event\_State = HIGH\_LIMIT

17. IF (the referenced property is writable) THEN

WRITE (referenced property) = (a value x:

(Setpoint\_Reference - Low\_Diff\_Limit + Deadband) < x < (Setpoint\_Reference + High\_Diff\_Limit – Deadband))

ELSE

MAKE (the referenced property have a value x:

(Setpoint\_Reference - Low\_Diff\_Limit + Deadband) < x < (Setpoint\_Reference + High\_Diff\_Limit – Deadband))

18. WAIT (Time\_Delay)

19. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the Loop object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-NORMAL transition),

‘Event Type’ = FLOATING\_LIMIT,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = HIGH\_LIMIT,

‘To State’ = NORMAL,

‘Event Values’ = reference-value, Status\_Flags, setpoint-value, error-limit,

20. TRANSMIT BACnet-SimpleACK-PDU

21. VERIFY STATUS\_FLAGS = (FALSE, FALSE, ?, ?)

22. VERIFY Event\_State = NORMAL

23. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

VERIFY Event\_Time\_Stamps = (the timestamp in step 8, \*, the timestamp in step 19)

24. IF (the referenced property is writable) THEN

WRITE (referenced property) = (a value x:

(Setpoint\_Reference - Low\_Diff\_Limit < x < (Setpoint\_Reference - Low\_Diff\_Limit + Deadband))

ELSE

MAKE (the referenced property have a value x:

(Setpoint\_Reference - Low\_Diff\_Limit < x < (Setpoint\_Reference - Low\_Diff\_Limit + Deadband))

25. WAIT (Time\_Delay + **Notification Fail Time**)

26. CHECK (verify that no notification message has been transmitted)

27. VERIFY Event\_State = NORMAL

28. IF (the referenced property is writable) THEN

WRITE (referenced property) = (a value x such x < (Setpoint\_Reference - Low\_Diff\_Limit))

ELSE

MAKE (referenced property have a value x: x < (Setpoint\_Reference - Low\_Diff\_Limit))

29. WAIT (Time\_Delay)

30. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the Loop object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-OFFNORMAL transition),

‘Event Type’ = FLOATING\_LIMIT,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = LOW\_LIMIT,

‘ Event Values’ = reference-value, Status\_Flags, setpoint-value, error-limit,

31. TRANSMIT BACnet-SimpleACK-PDU

32. VERIFY STATUS\_FLAGS = (TRUE, FALSE, ?, ?)

33. VERIFY Event\_State = LOW\_LIMIT

34. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

VERIFY Event\_Time\_Stamps = (the timestamp in step 30, \*, the timestamp in step 19)

35. IF (the referenced property is writable) THEN

WRITE (referenced property) = (a value x:

(Setpoint\_Reference - Low\_Limit) < x < (Setpoint\_Reference - Low\_Limit + Deadband))

ELSE

MAKE (the referenced property have a value x:

(Setpoint\_Reference - Low\_Limit) < x < (Setpoint\_Reference - Low\_Limit + Deadband))

36. WAIT (Time\_Delay + **Notification Fail Time**)

37. CHECK (verify that no notification message has been transmitted)

38. VERIFY Event\_State = Low\_Limit

39. IF (the referenced property is writable) THEN

WRITE (referenced property) = (a value x:

(Setpoint\_Reference - Low\_Limit + Deadband) < x < (Setpoint\_Reference + High\_Diff\_Limit – Deadband))

ELSE

MAKE (the referenced property have a value x:

(Setpoint\_Reference - Low\_Limit + Deadband) < x < (Setpoint\_Reference + High\_Diff\_Limit – Deadband))

40. WAIT (Time\_Delay)

41. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the Loop object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-NORMAL transition),

‘Event Type’ = FLOATING\_LIMIT,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = LOW\_LIMIT,

‘To State’ = NORMAL,

‘ Event Values’ = reference-value, Status\_Flags, setpoint-value, error-limit,

42. TRANSMIT BACnet-SimpleACK-PDU

43. VERIFY STATUS\_FLAGS = (FALSE, FALSE, ?, ?)

44. VERIFY Event\_State = NORMAL

45. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

VERIFY Event\_Time\_Stamps = (the timestamp in step 30, \*, the timestamp in step 41)

注意事项：测试描述中省略了Message Text参数，因为它是可选的。 IUT报告消息中可能包括该参数。在步骤12，23，34和45中由“\*”指示的时间戳可表示在步骤8的时间戳之前的时间值或未指定时间值。

1. OUT\_OF\_RANGE测试

所需测试：ReadProperty服务执行测试，10.18;WriteProperty服务执行测试，10.22。

BACnet参考条款：12.2,12.3,12.4,12.12,12.23,13.2,13.3.6和13.8。

目的：验证OUT\_OF\_RANGE事件算法。此测试适用于Event\_Type为OUT\_OF\_RANGE的Event Enrollment对象以及模拟输入，模拟输出和模拟值对象的内部事件报告。

测试方案：对象初始态为NORMAL。Present\_Value（参考属性）设置为较低但处于死区上限内的值，此时对象仍应处于NORMAL状态。Present\_Value（参考属性）设置为高于上限的值，在时间延迟到期后，对象进入HIGH\_LIMIT状态并发送事件通知消息。Present\_Value（参考属性）降低到低于上限但仍在限制的死区范围内的值，对象保持HIGH\_LIMIT状态。将Present\_Value（参考属性）进一步设置到不在限制死区范围内的正常值，延迟到期后，对象进入NORMAL状态并发出事件通知。重复相同的过程以测试出下限。

配置要求：配置IUT，使Event\_Enable属性值为TRUE（用于TO-OFFNORMAL和TO-NORMAL转换）；使用内部报告的对象，对于HighLimit和LowLimit事件，Limit\_Enable属性值为TRUE；Issue\_Confirmed\_Notifications属性值为TRUE；事件生成对象初始态为NORMAL。

在下述测试中，Present\_Value为参考属性。若测试Event Enrollment对象，则应使用适当的参考属性替换Present\_Value。

测试步骤：

1. VERIFY Event\_State = NORMAL

2. IF (Present\_Value is writable) THEN

WRITE Present\_Value = (a value x: (High\_Limit – Deadband)< x < High\_Limit)

ELSE

MAKE (Present\_Value have a value x: (High\_Limit – Deadband)< x < High\_Limit)

3. WAIT (Time\_Delay + **Notification Fail Time**)

4. CHECK (verify that no notification message has been transmitted)

5. VERIFY Event\_State = NORMAL

6. IF (Present\_Value is writable) THEN

WRITE Present\_Value = (a value x such x > High\_Limit)

ELSE

MAKE (Present\_Value have a value x: x > High\_Limit)

7. WAIT (Time\_Delay)

8. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-OFFNORMAL transition),

‘Event Type’ = OUT\_OF\_RANGE,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = HIGH\_LIMIT,

‘Event Values’ = Present\_Value, Status\_Flags, Deadband, High\_Limit

9. TRANSMIT BACnet-SimpleACK-PDU

10. VERIFY STATUS\_FLAGS = (TRUE, FALSE, ?, ?)

11. VERIFY Event\_State = HIGH\_LIMIT

12. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

VERIFY Event\_Time\_Stamps = (the timestamp in step 8, \*, \*)

13. IF (Present\_Value is writable) THEN

WRITE Present\_Value = (a value x: (High\_Limit – Deadband)< x < High\_Limit)

ELSE

MAKE (Present\_Value have a value x: (High\_Limit – Deadband)< x < High\_Limit)

14. WAIT (Time\_Delay + **Notification Fail Time**)

15. CHECK (verify that no notification message has been transmitted)

16. VERIFY Event\_State = HIGH\_LIMIT

17. IF (Present\_Value is writable) THEN

WRITE Present\_Value = (a value x: (Low\_Limit + Deadband) < x < (High\_Limit – Deadband))

ELSE

MAKE (Present\_Value have a value x: (Low\_Limit + Deadband) < x < (High\_Limit – Deadband))

18. WAIT (Time\_Delay)

19. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-NORMAL transition),

‘Event Type’ = OUT\_OF\_RANGE,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = HIGH\_LIMIT,

‘To State’ = NORMAL,

‘Event Values’ = Present\_Value, Status\_Flags, Deadband, High\_Limit

20. TRANSMIT BACnet-SimpleACK-PDU

21. VERIFY STATUS\_FLAGS = (FALSE, FALSE, ?, ?)

22. VERIFY Event\_State = NORMAL

23. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

VERIFY Event\_Time\_Stamps = (the timestamp in step 8, \*, the timestamp in step 19)

24. IF (Present\_Value is writable) THEN

WRITE Present\_Value = (a value x: Low\_Limit < x < (Low\_Limit + Deadband))

ELSE

MAKE (Present\_Value have a value x: Low\_Limit < x < (Low\_Limit + Deadband))

25. WAIT (Time\_Delay + **Notification Fail Time**)

26. CHECK (verify that no notification message has been transmitted)

27. VERIFY Event\_State = NORMAL

28. IF (Present\_Value is writable) THEN

WRITE Present\_Value = (a value x such x < Low\_Limit)

ELSE

MAKE (Present\_Value have a value x: x < Low\_Limit)

29. WAIT (Time\_Delay)

30. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-OFFNORMAL transition),

‘Event Type’ = OUT\_OF\_RANGE,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = LOW\_LIMIT,

‘Event Values’ = Present\_Value, Status\_Flags, Deadband, Low\_Limit

31. TRANSMIT BACnet-SimpleACK-PDU

32. VERIFY STATUS\_FLAGS = (TRUE, FALSE, ?, ?)

33. VERIFY Event\_State = LOW\_LIMIT

34. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

VERIFY Event\_Time\_Stamps = (the timestamp in step 30, \*, the timestamp in step 19)

35. IF (Present\_Value is writable) THEN

WRITE Present\_Value = (a value x: Low\_Limit < x < (Low\_Limit + Deadband))

ELSE

MAKE (Present\_Value have a value x: Low\_Limit < x < (Low\_Limit + Deadband))

36. WAIT (Time\_Delay + **Notification Fail Time**)

37. CHECK (verify that no notification message has been transmitted)

38. VERIFY Event\_State = LOW\_LIMIT

39. IF (Present\_Value is writable) THEN

WRITE Present\_Value = (a value x: (Low\_Limit + Deadband) < x < (High\_Limit – Deadband))

ELSE

MAKE (Present\_Value have a value x: (Low\_Limit + Deadband) < x < (High\_Limit – Deadband))

40. WAIT (Time\_Delay)

41. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-NORMAL transition),

‘Event Type’ = OUT\_OF\_RANGE,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = LOW\_LIMIT,

‘To State’ = NORMAL,

‘Event Values’ = Present\_Value, Status\_Flags, Deadband, Low\_Limit

42. TRANSMIT BACnet-SimpleACK-PDU

43. VERIFY STATUS\_FLAGS = (FALSE, FALSE, ?, ?)

44. VERIFY Event\_State = NORMAL

45. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

VERIFY Event\_Time\_Stamps = (the timestamp in step 30, \*, the timestamp in step 41)

注意事项：测试描述中省略了Message Text参数，因为它是可选的。 IUT报告消息中可能包括该参数。在步骤12，23，34和45中由“\*”指示的时间戳可表示在步骤8的时间戳之前的时间值或未指定时间值。

1. BUFFER\_READY测试

所需测试：ReadProperty服务执行测试，10.18。

BACnet参考条款：12.12,12.25,13.2,13.3.7和13.8。

目的：验证BUFFER\_READY事件算法操作正确。此测试适用于支持内部通知的趋势日志对象以及Event\_Type为BUFFER\_READY的Event Enrollment对象。

测试方案：执行报告的对象（“报告对象”）初始态为NORMAL，包含缓冲区的对象（“缓冲区对象”）中没有记录存储。缓冲区对象获取Records\_Since\_Notification指定的记录数，此时通知对象执行TO-NORMAL转换并发送BUFFER\_READY通知。

配置要求：配置IUT，Event\_Enable属性值为TRUE（用于TO\_ NORMAL转换）；测试开始时通知对象初始态为NORMAL；引用IUT的通知类Recipient\_List属性元素的“Issue Confirmed Notifications”参数应设置为TRUE。

测试步骤：

1. VERIFY Event\_State = NORMAL

2. MAKE (buffer object collect number of records specified by Notification\_Threshold)

3. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment

object being tested),

‘Time Stamp’ = (any appropriate BACnetTimeStamp value),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-NORMAL transition),

‘Event Type’ = BUFFER\_READY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = NORMAL,

‘Event Values’ = (BACnetObjectIdentifier of the IUT’s Device object),

(BACnetObjectIdentifier of the buffer object),

(any BACnetDateTime),

(current local BACnetDateTime)

4. TRANSMIT BACnet-SimpleACK-PDU

5. MAKE (buffer object collect number of records specified by Notification\_Threshold)

6. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (any appropriate BACnetTimeStamp value),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-NORMAL transition),

‘Event Type’ = BUFFER\_READY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = NORMAL,

‘Event Values’ = (BACnetObjectIdentifier of the IUT’s Device object),

(BACnetObjectIdentifier of the buffer object),

(current local BACnetDateTime sent in step 3),

(current local BACnetDateTime)

7. TRANSMIT BACnet-SimpleACK-PDU

1. CHANGE\_OF\_LIFE\_SAFETY测试
2. NORMAL到OFFNORMAL的转换测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.7,12.15,12.16,13.3.8,13.8和图13-9。

目的：此测试用例验证在NORMAL和OFFNORMAL事件状态之间转换的对象的CHANGE\_OF\_LIFE\_SAFETY算法。适用于Event\_Type为CHANGE\_OF\_LIFE\_SAFETY的Event Enrollment对象以及生命安全点和生命安全区对象的内部事件报告。

测试方案：对象初始态为NORMAL。Present\_Value（参考属性）更改为List\_Of\_Alarm\_Values中指定的值之一，延迟到期后，对象应进入OFFNORMAL状态并发送事件通知消息。

配置要求：配置IUT，使Event\_Enable属性值为TRUE（用于TO-OFFNORMAL，TO-FAULT和TO-NORMAL转换）Issue\_Confirmed\_Notifications属性值为TRUE；事件生成对象初始态为NORMAL。

在下述测试，Present\_Value为参考属性。若测试Event Enrollment对象，则应使用相应的参考属性替换Present\_Value。

测试步骤：

1. VERIFY Event\_State = NORMAL

2. MAKE (Present\_Value have a value x such that x corresponds to an OFFNORMAL state)

3. WAIT Time\_Delay

4. BEFORE **Notification Fail Time**

5. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-OFFNORMAL transition),

‘Event Type’ = CHANGE\_OF\_LIFE\_SAFETY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = OFFNORMAL,

‘Event Values’ = Present\_Value,Mode,Status\_Flags, Operation\_Expected

6. TRANSMIT BACnet-SimpleACK-PDU

7. VERIFY STATUS\_FLAGS = (TRUE, FALSE, ?, ?)

8. VERIFY Event\_State = OFFNORMAL

9. VERIFY Event\_Time\_Stamps = (the timestamp in step 5, \*, \*)

1. OFFNORMAL到NORMAL转换测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.7,12.15,12.16,13.3.8,13.8和图13-9。

目的：此测试用例验证在NORMAL和OFFNORMAL事件状态之间转换的对象的CHANGE\_OF\_LIFE\_SAFETY算法。适用于Event\_Type为CHANGE\_OF\_LIFE\_SAFETY的Event Enrollment对象以及生命安全点和生命安全区对象的内部事件报告。

测试方案：对象初始态为NORMAL。Present\_Value（参考属性）处于NORMAL状态。如果支持锁存，则对象应保持OFFNORMAL状态直到重置。需要对所有支持重置的操作重复此测试。如果不支持锁存，则在时间延迟到期后，对象应进入NORMAL状态并发送事件通知消息。TD通常能够通过使用具有重置或重置警报请求的LifeSafetyOperation服务来重置对象。

配置要求：配置IUT，使Event\_Enable属性值为TRUE（用于TO-OFFNORMAL，TO-FAULT和TO-NORMAL转换）。Issue\_Confirmed\_Notifications属性值为TRUE；事件生成对象初始态为OFFNORMAL。在下述测试中，Present\_Value为参考属性。若测试Event Enrollment对象，则应使用适当的参考属性替换Present\_Value。

测试步骤：

1. MAKE (Present\_Value have a value x such that x corresponds to a NORMAL state)

2. WAIT Time\_Delay

3. IF (latching is supported) THEN

4. CHECK (Event\_State = OFFNORMAL)

5. MAKE (the object reset)

6. BEFORE **Notification Fail Time**

7. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-OFFNORMAL transition),

‘Event Type’ = CHANGE\_OF\_LIFE\_SAFETY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = OFFNORMAL,

‘To State’ = NORMAL,

‘Event Values’ = Present\_Value, Mode, STATUS\_FLAGS, Operation\_Expected

8. TRANSMIT BACnet-SimpleACK-PDU,

9. VERIFY STATUS\_FLAGS = (FALSE, FALSE, ?, ?),

10. VERIFY Event\_State = NORMAL,

11. VERIFY Event\_Time\_Stamps = (the timestamp in step 7, \*, \*)

ELSE

12. BEFORE **Notification Fail Time**

13. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO- OFFNORMAL transition),

‘Event Type’ = CHANGE\_OF\_LIFE\_SAFETY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = OFFNORMAL,

‘To State’ = NORMAL,

‘Event Values’ = Present\_Value, Mode, STATUS\_FLAGS, Operation\_Expected

14. TRANSMIT BACnet-SimpleACK-PDU

15. VERIFY STATUS\_FLAGS = (FALSE, FALSE, ?, ?)

16. VERIFY Event\_State = NORMAL

17. VERIFY Event\_Time\_Stamps = (the timestamp in step 13, \*, \*)

1. NORMAL到LIFE\_SAFETY\_ALARM转换测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.7,12.15,12.16,13.3.8,13.8和图13-9。

目的：此测试用例验证在NORMAL和LIFE\_SAFETY\_ALARM事件状态之间转换的对象的CHANGE\_OF\_LIFE\_SAFETY算法。适用于Event\_Type为CHANGE\_OF\_LIFE\_SAFETY的Event Enrollment对象以及生命安全点和生命安全区对象的内部事件报告。

测试方案：对象初始态为NORMAL。Present\_Value（参考属性）更改为List\_Of\_Alarm\_Values中指定的值之一，延迟到期后，对象应进入LIFE\_SAFETY\_ALARM状态并发送事件通知消息。

配置要求：配置IUT，使Event\_Enable属性值为TRUE（用于TO-OFFNORMAL，TO-FAULT和TO-NORMAL转换）。Issue\_Confirmed\_Notifications属性值为TRUE；事件生成对象初始态为OFFNORMAL。

在下述测试中，Present\_Value为参考属性。若测试Event Enrollment对象，则应使用适当的参考属性替换Present\_Value。

测试步骤：

1. VERIFY Event\_State = NORMAL

2. MAKE (Present\_Value have a value x such that x corresponds to a LIFE\_SAFETY\_ALARM state)

3. WAIT Time\_Delay

4. BEFORE **Notification Fail Time**

5. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-OFFNORMAL transition),

‘Event Type’ = CHANGE\_OF\_LIFE\_SAFETY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = LIFE\_SAFETY\_ALARM,

‘ Event Values’ = Present\_Value, Mode, STATUS\_FLAGS, Operation\_Expected

6. TRANSMIT BACnet-SimpleACK-PDU

7. VERIFY STATUS\_FLAGS = (TRUE, FALSE, ?, ?)

8. VERIFY Event\_State = LIFE\_SAFETY\_ALARM

9. VERIFY Event\_Time\_Stamps = (the timestamp in step 5, \*, \*)

1. LIFE\_SAFETY\_ALARM到NORMAL转换测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.7,12.15,12.16,13.3.8,13.8和图13-9。

目的：此测试用例验证在NORMAL和LIFE\_SAFETY\_ALARM事件状态之间转换的对象的CHANGE\_OF\_LIFE\_SAFETY算法。适用于Event\_Type为CHANGE\_OF\_LIFE\_SAFETY的Event Enrollment对象以及生命安全点和生命安全区对象的内部事件报告。

测试方案：对象初始态为LIFE\_SAFETY\_ALARM。Present\_Value（参考属性）处于NORMAL状态。如果支持锁存，则对象应保持LIFE\_SAFETY\_ALARM，直到LifeSafetyOperation服务重置对象。需要对所有支持重置的操作重复此测试。如果不支持锁存，则在时间延迟到期后，对象应进入NORMAL状态并发送事件通知消息。

配置要求：配置IUT，使Event\_Enable属性值为TRUE（用于TO-OFFNORMAL，TO-FAULT和TO-NORMAL转换）。Issue\_Confirmed\_Notifications属性值为TRUE；事件生成对象初始态为LIFE\_SAFETY\_ALARM。在下述测试中，Present\_Value为参考属性。若测试Event Enrollment对象，则应使用适当的参考属性替换Present\_Value。

测试步骤：

1. MAKE (Present\_Value have a value x such that x corresponds to a NORMAL state)

2. WAIT Time\_Delay

3. IF (latching is supported) THEN

4. CHECK (Event\_State = LIFE\_SAFETY\_ALARM)

5. MAKE (the object reset)

6. BEFORE **Notification Fail Time**

7. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO- OFFNORMAL transition),

‘Event Type’ = CHANGE\_OF\_LIFE\_SAFETY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = LIFE\_SAFETY\_ALARM,

‘To State’ = NORMAL,

‘Event Values’ = Present\_Value, Mode, STATUS\_FLAGS, Operation\_Expected

8. TRANSMIT BACnet-SimpleACK-PDU

9. VERIFY STATUS\_FLAGS = (FALSE, FALSE, ?, ?)

10. VERIFY Event\_State = NORMAL

11. VERIFY Event\_Time\_Stamps = (the timestamp in step 7, \*, \*)

ELSE

12. BEFORE **Notification Fail Time**

13. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO- OFFNORMAL transition),

‘Event Type’ = CHANGE\_OF\_LIFE\_SAFETY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = LIFE\_SAFETY\_ALARM,

‘To State’ = NORMAL,

‘Event Values’ = Present\_Value, Mode, STATUS\_FLAGS, Operation\_Expected

14. TRANSMIT BACnet-SimpleACK-PDU

15. VERIFY STATUS\_FLAGS = (FALSE, FALSE, ?, ?)

16. VERIFY Event\_State = NORMAL

17. VERIFY Event\_Time\_Stamps = (the timestamp in step 13, \*, \*)

1. LIFE\_SAFETY\_ALARM到OFFNORMAL转换测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.7,12.15,12.16,13.3.8,13.8和图13-9。

目的：此测试用例验证在LIFE\_SAFETY\_ALARM和OFFNORMAL事件状态之间转换的对象的CHANGE\_OF\_LIFE\_SAFETY算法。适用于Event\_Type为CHANGE\_OF\_LIFE\_SAFETY的Event Enrollment对象以及生命安全点和生命安全区对象的内部事件报告。

测试方案：对象初始态为LIFE\_SAFETY\_ALARM。Present\_Value（参考属性）处于OFFNORMAL状态。如果支持锁存，则对象应保持LIFE\_SAFETY\_ALARM，直到LifeSafetyOperation服务重置对象。需要对所有支持重置的操作重复此测试。如果不支持锁存，则在时间延迟到期后，对象应进入OFFNORMAL状态并发送事件通知消息。

配置要求：配置IUT，使Event\_Enable属性值为TRUE（用于TO-OFFNORMAL，TO-FAULT和TO-NORMAL转换）。Issue\_Confirmed\_Notifications属性值为TRUE；事件生成对象初始态为LIFE\_SAFETY\_ALARM。在下述测试中，Present\_Value为参考属性。若测试Event Enrollment对象，则应使用适当的参考属性替换Present\_Value。TD通常能够通过使用具有重置或重置警报请求的LifeSafetyOperation服务来重置对象。

测试步骤：

1. MAKE (Present\_Value have a value x such that x corresponds to an OFFNORMAL state)

2. WAIT Time\_Delay

3. IF (latching is supported) THEN

4. CHECK (Event\_State = OFFNORMAL)

5. MAKE (the object reset)

6. BEFORE **Notification Fail Time**

7. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-OFFNORMAL transition),

‘Event Type’ = CHANGE\_OF\_LIFE\_SAFETY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = LIFE\_SAFETY\_ALARM,

‘To State’ = OFFNORMAL,

‘Event Values’ = Present\_Value, Mode, STATUS\_FLAGS, Operation\_Expected

8. TRANSMIT BACnet-SimpleACK-PDU

9. VERIFY STATUS\_FLAGS = (FALSE, FALSE, ?, ?)

10. VERIFY Event\_State = OFFNORMAL

11. VERIFY Event\_Time\_Stamps = (the timestamp in step 7, \*, \*)

ELSE

12. BEFORE **Notification Fail Time**

13. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO- OFFNORMAL transition),

‘Event Type’ = CHANGE\_OF\_LIFE\_SAFETY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = LIFE\_SAFETY\_ALARM,

‘To State’ = OFFNORMAL,

‘Event Values’ = Present\_Value, Mode, STATUS\_FLAGS, Operation\_Expected

14. TRANSMIT BACnet-SimpleACK-PDU

15. VERIFY STATUS\_FLAGS = (FALSE, FALSE, ?, ?)

16. VERIFY Event\_State = OFFNORMAL

17. VERIFY Event\_Time\_Stamps = (the timestamp in step 13, \*, \*)

1. OFFNORMAL到LIFE\_SAFETY\_ALARM转换测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.7,12.15,12.16,13.3.8,13.8和图13-9。

目的：此测试用例验证在OFFNORMAL和LIFE\_SAFETY\_ALARM事件状态之间转换的对象的CHANGE\_OF\_LIFE\_SAFETY算法。适用于Event\_Type为CHANGE\_OF\_LIFE\_SAFETY的Event Enrollment对象以及生命安全点和生命安全区对象的内部事件报告。

测试方案：对象初始态为OFFNORMAL。Present\_Value（参考属性）处于LIFE\_SAFETY\_ALARM状态。如果支持锁存，则对象应保持OFFNORMAL，直到LifeSafetyOperation服务重置对象。需要对所有支持重置的操作重复此测试。如果不支持锁存，则在时间延迟到期后，对象应进入LIFE\_SAFETY\_ALARM状态并发送事件通知消息。

配置要求：配置IUT，使Event\_Enable属性值为TRUE（用于TO-OFFNORMAL，TO-FAULT和TO-NORMAL转换）。Issue\_Confirmed\_Notifications属性值为TRUE；事件生成对象初始态为OFFNORMAL。在下述测试中，Present\_Value为参考属性。若测试Event Enrollment对象，则应使用适当的参考属性替换Present\_Value。TD通常能够通过使用具有重置或重置警报请求的LifeSafetyOperation服务来重置对象。

测试步骤：

1. MAKE (Present\_Value have a value x such that x corresponds to an LIFE\_SAFETY state)

2. WAIT Time\_Delay

3. IF (latching is supported) THEN

4. CHECK (Event\_State = LIFE\_SAFETY\_ALARM)

5. MAKE (the object reset)

6. BEFORE **Notification Fail Time**

7. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO- OFFNORMAL transition),

‘Event Type’ = CHANGE\_OF\_LIFE\_SAFETY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = OFFNORMAL,

‘To State’ = LIFE\_SAFETY\_ALARM,

‘Event Values’ = Present\_Value, Mode, STATUS\_FLAGS, Operation\_Expected

8. TRANSMIT BACnet-SimpleACK-PDU

9. VERIFY STATUS\_FLAGS = (FALSE, FALSE, ?, ?)

10. VERIFY Event\_State = LIFE\_SAFETY\_ALARM

11. VERIFY Event\_Time\_Stamps = (the timestamp in step 7, \*, \*)

ELSE

12. BEFORE **Notification Fail Time**

13. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO- OFFNORMAL transition),

‘Event Type’ = CHANGE\_OF\_LIFE\_SAFETY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = OFFNORMAL,

‘To State’ = LIFE\_SAFETY\_ALARM,

‘Event Values’ = Present\_Value, Mode, STATUS\_FLAGS, Operation\_Expected

14. TRANSMIT BACnet-SimpleACK-PDU

15. VERIFY STATUS\_FLAGS = (FALSE, FALSE, ?, ?)

16. VERIFY Event\_State = LIFE\_SAFETY\_ALARM

17. VERIFY Event\_Time\_Stamps = (the timestamp in step 13, \*, \*)

1. 事件状态保持时的模式转换测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.7,12.15,12.16,13.3.8,13.8和图13-9。

目的：此测试用例验证发生模式更改时在NORMAL，OFFNORMAL和LIFE\_SAFETY\_ALARM事件状态之间转换的对象的CHANGE\_OF\_LIFE\_SAFETY算法。当模式更改但事件状态不更改时执行测试，当模式改变与事件状态改变同时发生时也进行测试。在后一种情况下，测试验证通知立即进行而不等待时间延迟。测试适用于Event\_Type为CHANGE\_OF\_LIFE\_SAFETY的Event Enrollment对象以及生命安全点和生命安全区对象的内部事件报告。

测试方案：对象初始态为NORMAL。改变模式，时间延迟到期后，对象发送事件通知消息，在初态为OFFNORMAL和LIFE\_SAFETY\_ALARM时也进行此操作。

然后通过更改Mode属性并选择List\_Of\_Alarm\_Values中指定的Present\_Value，重复测试，对象应立即进入OFFNORMAL状态并发送事件通知消息。然后将Present\_Value（参考属性）更改为与NORMAL状态对应的值，同时写入Mode。对象应立即进入NORMAL状态并发送事件通知消息。

配置要求：配置IUT，使Event\_Enable属性值为TRUE（用于TO-OFFNORMAL，TO-FAULT和TO-NORMAL转换）。Issue\_Confirmed\_Notifications属性值为TRUE；事件生成对象初始态为NORMAL。

在下述测试中，Present\_Value为参考属性。若测试Event Enrollment对象，则应使用适当的参考属性替换Present\_Value。

测试步骤：

1. CHECK (Event\_State = NORMAL)

2. MAKE (Present\_Value have a value x such that x corresponds to a NORMAL state)

3. BEFORE **Notification Fail Time**

4. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-NORMAL transition),

‘Event Type’ = CHANGE\_OF\_LIFE\_SAFETY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = NORMAL,

‘Event Values’ = Present\_Value, Mode, STATUS\_FLAGS, Operation\_Expected

5. TRANSMIT BACnet-SimpleACK-PDU

6. VERIFY STATUS\_FLAGS = (FALSE, FALSE, ?, ?)

7. VERIFY Event\_State = NORMAL

8. VERIFY Event\_Time\_Stamps = (the timestamp in step 4, \*, \*)

9. MAKE (Present\_Value have a value x such that x corresponds to an OFFNORMAL state)

10. MAKE (Mode = different value that maintains Event\_State as OFFNORMAL)

11. BEFORE **Notification Fail Time**

12. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-OFFNORMAL transition),

‘Event Type’ = CHANGE\_OF\_LIFE\_SAFETY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = OFFNORMAL,

‘To State’ = OFFNORMAL,

‘Event Values’ = Present\_Value, Mode, STATUS\_FLAGS, Operation\_Expected

13. TRANSMIT BACnet-SimpleACK-PDU

14. VERIFY STATUS\_FLAGS = (TRUE, FALSE, ?, ?)

15. VERIFY Event\_State = OFFNORMAL

16. VERIFY Event\_Time\_Stamps = (the timestamp in step 12, \*, \*)

17. MAKE (Present\_Value have a value x such that x corresponds to a LIFE\_SAFETY\_ALARM state)

18. MAKE (Mode = different value that maintains Event\_State = LIFE\_SAFETY\_ALARM)

19. BEFORE **Notification Fail Time**

20. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-NORMAL transition),

‘Event Type’ = CHANGE\_OF\_LIFE\_SAFETY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = OFFNORMAL,

‘To State’ = OFFNORMAL,

‘Event Values’ = Present\_Value, Mode, STATUS\_FLAGS, Operation\_Expected

21. TRANSMIT BACnet-SimpleACK-PDU

22. VERIFY STATUS\_FLAGS = (TRUE, FALSE, ?, ?)

23. VERIFY Event\_State = OFFNORMAL

24. VERIFY Event\_Time\_Stamps = (the timestamp in step 20, \*, \*)

1. NORMAL到OFFNORMAL模式转换测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.7,12.15,12.16,13.3.8,13.8和图13-9。

目的：此测试用例验证在NORMAL和OFFNORMAL事件状态之间转换的对象的CHANGE\_OF\_LIFE\_SAFETY算法。适用于Event\_Type为CHANGE\_OF\_LIFE\_SAFETY的Event Enrollment对象以及生命安全点和生命安全区对象的内部事件报告。

测试方案：对象初始态为NORMAL。更改Mode属性为使对象进入OFFNORMAL状态的值。对象应立即进入OFFNORMAL状态并发送事件通知消息。

配置要求：配置IUT，使Event\_Enable属性值为TRUE（用于TO-OFFNORMAL，TO-FAULT和TO-NORMAL转换）。Issue\_Confirmed\_Notifications属性值为TRUE；事件生成对象初始态为NORMAL。

在下述测试中，Present\_Value为参考属性。若测试Event Enrollment对象，则应使用适当的参考属性替换Present\_Value。

测试步骤：

1. VERIFY Event\_State = NORMAL

2. MAKE (Mode a different value that forces the state to OFFNORMAL)

3. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-OFFNORMAL transition),

‘Event Type’ = CHANGE\_OF\_LIFE\_SAFETY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = OFFNORMAL,

‘Event Values’ = Present\_Value, Mode, STATUS\_FLAGS, Operation\_Expected

4. TRANSMIT BACnet-SimpleACK-PDU

5. VERIFY STATUS\_FLAGS = (TRUE, FALSE, ?, ?)

6. VERIFY Event\_State = OFFNORMAL

7. VERIFY Event\_Time\_Stamps = (the timestamp in step 3, \*, \*)

1. OFFNORMAL到NORMAL模式转换测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.7,12.15,12.16,13.3.8,13.8和图13-9。

目的：此测试用例验证在OFFNORMAL和NORMAL事件状态之间转换的对象的CHANGE\_OF\_LIFE\_SAFETY算法。适用于Event\_Type为CHANGE\_OF\_LIFE\_SAFETY的Event Enrollment对象以及生命安全点和生命安全区对象的内部事件报告。

测试方案：对象初始态为OFFNORMAL。更改Mode属性为使对象进入NORMAL状态的值。如果支持锁存，则对象应保持OFFNORMAL，直到LifeSafetyOperation服务重置对象。需要对所有支持重置的操作重复此测试。如果不支持锁存，则在时间延迟到期后，对象应进入NORMAL状态并发送事件通知消息。

配置要求：配置IUT，使Event\_Enable属性值为TRUE（用于TO-OFFNORMAL，TO-FAULT和TO-NORMAL转换）。Issue\_Confirmed\_Notifications属性值为TRUE；事件生成对象初始态为OFFNORMAL。在下述测试中，Present\_Value为参考属性。若测试Event Enrollment对象，则应使用适当的参考属性替换Present\_Value。TD通常能够通过使用具有重置或重置警报请求的LifeSafetyOperation服务来重置对象。

测试步骤：

1. MAKE (Mode a different value that forces the state to NORMAL)

2. IF (latching is supported) THEN

3. CHECK (Event\_State = OFFNORMAL)

4. MAKE (the object reset)

5. BEFORE **Notification Fail Time**

6. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-OFFNORMAL transition),

‘Event Type’ = CHANGE\_OF\_LIFE\_SAFETY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = OFFNORMAL,

‘To State’ = NORMAL,

‘Event Values’ = Present\_Value, Mode, STATUS\_FLAGS, Operation\_Expected

7. TRANSMIT BACnet-SimpleACK-PDU

8. VERIFY STATUS\_FLAGS = (FALSE, FALSE, ?, ?)

9. VERIFY Event\_State = NORMAL

10. VERIFY Event\_Time\_Stamps = (the timestamp in step 6, \*, \*)

ELSE

11. BEFORE **Notification Fail Time**

12. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO- OFFNORMAL transition),

‘Event Type’ = CHANGE\_OF\_LIFE\_SAFETY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = OFFNORMAL,

‘To State’ = NORMAL,

‘Event Values’ = Present\_Value, Mode, STATUS\_FLAGS, Operation\_Expected

13. TRANSMIT BACnet-SimpleACK-PDU

14. VERIFY STATUS\_FLAGS = (FALSE, FALSE, ?, ?)

15. VERIFY Event\_State = NORMAL

16. VERIFY Event\_Time\_Stamps = (the timestamp in step 12, \*,\*)

1. NORMAL到LIFE\_SAFETY\_ALARM模式转换测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.7,12.15,12.16,13.3.8,13.8和图13-9。

目的：此测试用例验证改变状态在NORMAL和LIFE\_SAFETY\_ALARM事件状态之间转换的对象的CHANGE\_OF\_LIFE\_SAFETY算法。适用于Event\_Type为CHANGE\_OF\_LIFE\_SAFETY的Event Enrollment对象以及生命安全点和生命安全区对象的内部事件报告。

测试方案：对象初始态为NORMAL。更改Mode属性为使对象进入LIFE\_SAFETY\_ALARM状态的值，对象应立即进入LIFE\_SAFETY\_ALARM状态并发送事件通知消息。

在下述测试中，Present\_Value为参考属性。若测试Event Enrollment对象，则应使用适当的参考属性替换Present\_Value。

测试步骤：

1. VERIFY Event\_State = NORMAL

2. MAKE (Mode a different value that forces the state to LIFE\_SAFETY\_ALARM)

3. BEFORE **Notification Fail Time**

4. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-OFFNORMAL transition),

‘Event Type’ = CHANGE\_OF\_LIFE\_SAFETY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = LIFE\_SAFETY\_ALARM,

‘Event Values’ = Present\_Value, Mode, STATUS\_FLAGS, Operation\_Expected

5. TRANSMIT BACnet-SimpleACK-PDU

6. VERIFY STATUS\_FLAGS = (TRUE, FALSE, ?, ?)

7. VERIFY Event\_State = LIFE\_SAFETY\_ALARM

8. VERIFY Event\_Time\_Stamps = (the timestamp in step 4, \*, \*)

1. LIFE\_SAFETY\_ALARM到NORMAL模式转换测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.7,12.15,12.16,13.3.8,13.8和图13-9。

目的：此测试用例验证在LIFE\_SAFETY\_ALARM和NORMAL事件状态之间转换的对象的CHANGE\_OF\_LIFE\_SAFETY算法。适用于Event\_Type为CHANGE\_OF\_LIFE\_SAFETY的Event Enrollment对象以及生命安全点和生命安全区对象的内部事件报告。

测试方案：对象初始态为LIFE\_SAFETY\_ALARM。更改Mode属性为使对象进入NORMAL状态的值。如果支持锁存，则对象应保持LIFE\_SAFETY\_ALARM，直到LifeSafetyOperation服务重置对象。需要对所有支持重置的操作重复此测试。如果不支持锁存，则在时间延迟到期后，对象应进入NORMAL状态并发送事件通知消息。

配置要求：配置IUT，使Event\_Enable属性值为TRUE（用于TO-OFFNORMAL，TO-FAULT和TO-NORMAL转换）。Issue\_Confirmed\_Notifications属性值为TRUE；事件生成对象初始态为LIFE\_SAFETY\_ALARM。在下述测试中，Present\_Value为参考属性。若测试Event Enrollment对象，则应使用适当的参考属性替换Present\_Value。TD通常能够通过使用具有重置或重置警报请求的LifeSafetyOperation服务来重置对象。

测试步骤：

1. MAKE (Mode a different value that forces the state to NORMAL)

2. IF (latching is supported) THEN

3. CHECK (Event\_State = LIFE\_SAFETY\_ALARM)

4. MAKE (the object reset)

5. BEFORE **Notification Fail Time**

6. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-OFFNORMAL transition),

‘Event Type’ = CHANGE\_OF\_LIFE\_SAFETY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = LIFE\_SAFETY\_ALARM,

‘To State’ = NORMAL,

‘Event Values’ = Present\_Value, Mode, STATUS\_FLAGS, Operation\_Expected

7. TRANSMIT BACnet-SimpleACK-PDU

8. VERIFY STATUS\_FLAGS = (FALSE, FALSE, ?, ?)

9. VERIFY Event\_State = NORMAL

10. VERIFY Event\_Time\_Stamps = (the timestamp in step 6, \*, \*)

ELSE

11. BEFORE **Notification Fail Time**

12. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-OFFNORMAL transition),

‘Event Type’ = CHANGE\_OF\_LIFE\_SAFETY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = LIFE\_SAFETY\_ALARM,

‘To State’ = NORMAL,

‘Event Values’ = Present\_Value, Mode, STATUS\_FLAGS, Operation\_Expected

13. TRANSMIT BACnet-SimpleACK-PDU

14. VERIFY STATUS\_FLAGS = (FALSE, FALSE, ?, ?)

15. VERIFY Event\_State = NORMAL

16. VERIFY Event\_Time\_Stamps = (the timestamp in step 12, \*, \*)}

1. LIFE\_SAFETY\_ALARM到 OFFNORMAL模式转换测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.7,12.15,12.16,13.3.8,13.8和图13-9。

目的：此测试用例验证在LIFE\_SAFETY\_ALARM和OFFNORMAL事件状态之间转换的对象的CHANGE\_OF\_LIFE\_SAFETY算法。适用于Event\_Type为CHANGE\_OF\_LIFE\_SAFETY的Event Enrollment对象以及生命安全点和生命安全区对象的内部事件报告。

测试方案：对象初始态为LIFE\_SAFETY\_ALARM。更改Mode属性为使对象进入OFFNORMAL状态的值。如果支持锁存，则对象应保持LIFE\_SAFETY\_ALARM，直到LifeSafetyOperation服务重置对象。需要对所有支持重置的操作重复此测试。如果不支持锁存，则在时间延迟到期后，对象应进入OFFNORMAL状态并发送事件通知消息。

配置要求：配置IUT，使Event\_Enable属性值为TRUE（用于TO-OFFNORMAL，TO-FAULT和TO-NORMAL转换）。Issue\_Confirmed\_Notifications属性值为TRUE；事件生成对象初始态为LIFE\_SAFETY\_ALARM。在下述测试中，Present\_Value为参考属性。若测试Event Enrollment对象，则应使用适当的参考属性替换Present\_Value。TD通常能够通过使用具有重置或重置警报请求的LifeSafetyOperation服务来重置对象。

测试步骤：

1. MAKE (Mode a different value that forces the state to OFFNORMAL)

2. IF (latching is supported) THEN

3. CHECK (Event\_State = OFFNORMAL)

4. MAKE (the object reset)

5. BEFORE **Notification Fail Time**

6. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-OFFNORMAL transition),

‘Event Type’ = CHANGE\_OF\_LIFE\_SAFETY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = LIFE\_SAFETY\_ALARM,

‘To State’ = OFFNORMAL,

‘Event Values’ = Present\_Value, Mode, STATUS\_FLAGS, Operation\_Expected

7. TRANSMIT BACnet-SimpleACK-PDU

8. VERIFY STATUS\_FLAGS = (FALSE, FALSE, ?, ?)

9. VERIFY Event\_State = OFFNORMAL

10. VERIFY Event\_Time\_Stamps = (the timestamp in step 6, \*, \*)

ELSE

11. BEFORE **Notification Fail Time**

12. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO- OFFNORMAL transition),

‘Event Type’ = CHANGE\_OF\_LIFE\_SAFETY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = LIFE\_SAFETY\_ALARM,

‘To State’ = OFFNORMAL,

‘Event Values’ = Present\_Value, Mode, STATUS\_FLAGS, Operation\_Expected

13. TRANSMIT BACnet-SimpleACK-PDU

14. VERIFY STATUS\_FLAGS = (FALSE, FALSE, ?, ?)

15. VERIFY Event\_State = OFFNORMAL

16. VERIFY Event\_Time\_Stamps = (the timestamp in step 12, \*, \*)

1. OFFNORMAL到LIFE\_SAFETY\_ALARM模式转换测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.7,12.15,12.16,13.3.8,13.8和图13-9。

目的：此测试用例验证通过更改Mode属性使对象在OFFNORMAL和LIFE\_SAFETY\_ALARM 事件状态之间转换的CHANGE\_OF\_LIFE\_SAFETY算法操作。适用于Event\_Type为CHANGE\_OF\_LIFE\_SAFETY的Event Enrollment对象以及生命安全点和生命安全区对象的内部事件报告。

测试方案：对象初始态为OFFNORMAL。更改Mode属性为使对象进入LIFE\_SAFETY\_ALARM状态的值。如果支持锁存，则对象应保持OFFNORMAL，直到LifeSafetyOperation服务重置对象。需要对所有支持重置的操作重复此测试。如果不支持锁存，则在时间延迟到期后，对象应进入LIFE\_SAFETY\_ALARM状态并发送事件通知消息。

配置要求：配置IUT，使Event\_Enable属性值为TRUE（用于TO-OFFNORMAL，TO-FAULT和TO-NORMAL转换）。Issue\_Confirmed\_Notifications属性值为TRUE；事件生成对象初始态为LIFE\_SAFETY\_ALARM。在下述测试中，Present\_Value为参考属性。若测试Event Enrollment对象，则应使用适当的参考属性替换Present\_Value。TD通常能够通过使用具有重置或重置警报请求的LifeSafetyOperation服务来重置对象。

测试步骤：

1. MAKE (Mode a different value that forces the state to LIFE\_SAFETY\_ALARM)

2. IF (latching is supported) THEN

3. CHECK (Event\_State = OFFNORMAL)

4. MAKE (the object reset)

5. BEFORE **Notification Fail Time**

6. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO OFFNORMAL transition),

‘Event Type’ = CHANGE\_OF\_LIFE\_SAFETY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = LIFE\_SAFETY\_ALARM,

‘To State’ = OFFNORMAL,

‘Event Values’ = Present\_Value, Mode, STATUS\_FLAGS, Operation\_Expected

7. TRANSMIT BACnet-SimpleACK-PDU

8. VERIFY STATUS\_FLAGS = (FALSE, FALSE, ?, ?)

9. VERIFY Event\_State = OFFNORMAL

10. VERIFY Event\_Time\_Stamps = (the timestamp in step 6, \*, \*)

ELSE

11. BEFORE **Notification Fail Time**

12. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-OFFNORMAL transition),

‘Event Type’ = CHANGE\_OF\_LIFE\_SAFETY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = LIFE\_SAFETY\_ALARM,

‘To State’ = OFFNORMAL,

‘Event Values’ = Present\_Value, Mode, STATUS\_FLAGS, Operation\_Expected

13. TRANSMIT BACnet-SimpleACK-PDU

14. VERIFY STATUS\_FLAGS = (FALSE, FALSE, ?, ?)

15. VERIFY Event\_State = OFFNORMAL

16. VERIFY Event\_Time\_Stamps = (the timestamp in step 12, \*, \*)

1. TO-FAULT转变测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.7,12.15,12.16,13.3.8,13.8和图13-9。

目的：此测试用例验证发生模式更改时在NORMAL，FAULT事件状态之间转换的对象的CHANGE\_OF\_LIFE\_SAFETY算法。测试适用于Event\_Type为CHANGE\_OF\_LIFE\_SAFETY的Event Enrollment对象以及生命安全点和生命安全区对象的内部事件报告。

测试方案：对象初始态为NORMAL。Present\_Value（参考属性）更改为故障值列表中指定的值之一。 在时间延迟到期后，对象应进入FAULT状态并发送事件通知消息。该测试还包含从FAULT到NORMAL的返回转换改变模式。

配置要求：配置IUT，使Event\_Enable属性值为TRUE（用于TO-OFFNORMAL，TO-FAULT和TO-NORMAL转换）。Issue\_Confirmed\_Notifications属性值为TRUE；事件生成对象初始态为NORMAL。

在下述测试中，Present\_Value为参考属性。

测试步骤：

1. VERIFY Event\_State = NORMAL

2. MAKE (Present\_Value have a value x such that x corresponds to a FAULT state)

3. BEFORE **Notification Fail Time**

4. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-FAULT transition),

‘Event Type’ = CHANGE\_OF\_LIFE\_SAFETY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = FAULT,

‘Event Values’ = Present\_Value, Mode, STATUS\_FLAGS, Operation\_Expected

5. TRANSMIT BACnet-SimpleACK-PDU

6. VERIFY STATUS\_FLAGS = (TRUE, FALSE, ?, ?)

7. VERIFY Event\_State = FAULT

8. VERIFY Event\_Time\_Stamps = (the timestamp in step 5, \*, \*)

9. MAKE (Present\_Value have a value x such that x corresponds to a NORMAL state)

10. BEFORE **Notification Fail Time**

11. RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (the value configured to correspond to a TO-NORMAL transition),

‘Event Type’ = CHANGE\_OF\_LIFE\_SAFETY,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = FAULT,

‘To State’ = NORMAL,

‘Event Values’ = Present\_Value, Mode, STATUS\_FLAGS, Operation\_Expected

12. TRANSMIT BACnet-SimpleACK-PDU

13. VERIFY STATUS\_FLAGS = (TRUE, FALSE, ?, ?)

14. VERIFY Event\_State = NORMAL

15. VERIFY Event\_Time\_Stamps = (the timestamp in step 13, \*, \*)

1. EXTENDED测试

所需测试：无

BACnet参考条款：13.2和13.3。

目的：此测试验证扩展的ConfirmedEventNotification messages消息的正确生成。它适用于Event\_Type为EXTENDED的事件生成对象，这适用于使用专有算法的任何Event Enrollment对象。

测试方案：事件生成对象初始态为ONORMAL。事件生成对象可通过任意方式转换到任何状态，接收并验证生成的ConfirmedEventNotification消息。

配置要求：配置IUT，使测试中的全部转换的Event\_Enable属性值都为TRUE；Issue\_Confirmed\_Notifications属性值为TRUE。在测试开始时，事件发生对象应处于NORMAL状态。

测试步骤：

1. VERIFY Event\_State = NORMAL

2. MAKE (the event generating object transition)

3. WAIT (Time\_Delay)

4. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the event generating object being tested),

‘Time Stamp’ = (the current local time),

‘Notification Class’ = (the configured notification class, or absent if the event generating object is using a Recipient property instead),

‘Priority’ = (the value configured for this transition),

‘Event Type’ = EXTENDED,

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = NORMAL,

‘To State’ = (the state the object was made to transition to),

‘Event Values’ = VendorId, extendedEventType, and any other values as defined by the vendor

发送结果：测试描述中省略了Message Text参数，因为它是可选的，IUT通知消息中包括该参数。

1. 网络优先级测试

所需测试：ReadProperty服务执行测试，10.18;；WriteProperty服务执行测试，10.22。

BACnet参考条款：13.4.1

目的：验证在传输EventNotification请求时是否使用了正确的网络优先级。

测试方案：IUT生成具有特定优先级的ConfirmedEventNotification，检查NPCI中的网络优先级，确保其基于事件的优先级是正确的。

测试配置：IUT配置引用通知类对象N1的事件生成对象E1。

测试步骤：

1. IF (the IUT can generate ConfirmedEventNotifications with an event priority in the range 0..63) THEN

2. MAKE (The IUT initiate a ConfirmedEventNotification-Request with a event priority in the range 0..63)

3. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Network Priority’ = 3

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time of the IUT),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (a value in the range 0..63),

‘Event Type’ = (any valid value),

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = (any valid value),

‘To State’ = (any valid value),

‘ Event Values’ = (any valid valid)

4. TRANSMIT BACnet-SimpleACK-PDU

5. IF (the IUT can generate ConfirmedEventNotifications with an event priority in the range 64..127) THEN

6. MAKE (The IUT initiate a ConfirmedEventNotification-Request with an event priority in the range 64..127)

7. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Network Priority’ = 2

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time of the IUT),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (a value in the range 64..127),

‘Event Type’ = (any valid value),

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = (any valid value),

‘To State’ = (any valid value),

‘Event Values’ = (any valid valid)

8. TRANSMIT BACnet-SimpleACK-PDU

9. IF (the IUT can generate ConfirmedEventNotifications with an event priority in the range 128..191) THEN

10. MAKE (The IUT initiate a ConfirmedEventNotification-Request with a event priority in the range 128..191)

11. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Network Priority’ = 1

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time of the IUT),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (a value in the range 128..191),

‘Event Type’ = (any valid value),

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = (any valid value),

‘To State’ = (any valid value),

‘ Event Values’ = (any valid valid)

12. TRANSMIT BACnet-SimpleACK-PDU

13. IF (the IUT can generate ConfirmedEventNotifications with an event priority in the range 192..255)THEN

14. MAKE (The IUT initiate a ConfirmedEventNotification-Request with a event priority in the range 192..255)

15. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

‘Network Priority’ = 0

‘Process Identifier’ = (any valid process ID),

‘Initiating Device Identifier’ = IUT,

‘Event Object Identifier’ = (the intrinsic reporting object or the Event Enrollment object being tested),

‘Time Stamp’ = (the current local time of the IUT),

‘Notification Class’ = (the configured notification class),

‘Priority’ = (a value in the range 192..255),

‘Event Type’ = (any valid value),

‘Notify Type’ = EVENT | ALARM,

‘AckRequired’ = TRUE | FALSE,

‘From State’ = (any valid value),

‘To State’ = (any valid value),

‘ Event Values’ = (any valid valid)

16. TRANSMIT BACnet-SimpleACK-PDU

* 1. 未确认事件通知（UnconfirmedEventNotification）服务启动测试

本节定义了执行支持启动UnconfirmedEventNotification服务请求所必需的测试。UnconfirmedEventNotification测试针对特定事件检测算法使用。对于支持内部事件报告的对象类型，IUT应通过对适用于该对象类型的事件检测算法的测试。如果IUT支持Event Enrollment对象，它将通过对应于每个事件类型的事件检测算法的测试。

1. CHANGE\_OF\_BITSTRING位串变化测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12,13.3.1和13.9。

目的：验证位串变化事件算法。此测试适用于Event\_Type为CHANGE\_OF\_BITSTRING的Event Enrollment对象。

配置要求：配置IUT，使Event\_Enable属性值为TRUE（用于TO-OFFNORMAL，TO-FAULT和TO-NORMAL转换）；Issue\_Confirmed\_Notifications属性值为FALSE；事件生成对象初始态为NORMAL。

测试步骤：除使用UnconfirmedEventNotification服务请求代替ConfirmedEventNotification服务请求发送事件报告外，此测试用例的测试步骤与8.4.1中的测试步骤相同。

注意事项：除使用UnconfirmedEventNotification服务请求代替ConfirmedEventNotification服务请求发送事件报告外此测试用例的结果与8.4.1中的相同。用于这些消息的MAC地址应该是到达TD本地网络的广播，或者是TD的MAC地址。

1. CHANGE\_OF\_STATE状态变化测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.6, 12.8, 12.18, 12.20, 13.2,13.3.2和13.9。

目的：验证CHANGE\_OF\_STATE事件算法。此测试适用于Event\_Type为状态变化的Event Enrollment对象和二进制输入、二进制值、多状态输入和多状态值对象的内部事件报告。

配置要求：配置IUT，使Event\_Enable属性值为TRUE（用于TO-OFFNORMAL，TO-NORMAL转换）；Issue\_Confirmed\_Notifications属性值为FALSE；事件生成对象初始态为NORMAL。

测试步骤：除使用UnconfirmedEventNotification服务请求代替ConfirmedEventNotification服务请求发送事件报告和TD不收到通知外，此测试用例的测试步骤与9.4.2中的测试步骤相同。

注意事项：除使用未确认事件通知服务请求发送事件报告外，此测试用例的结果与9.4.2中的相同。用于这些消息的MAC地址应该是到达TD本地网络的广播，或者是TD的MAC地址。

1. CHANGE\_OF\_VALUE测试

本节定义了执行CHANGE\_OF\_VALUE事件算法所必需的测试。CHANGE\_OF\_VALUE算法可以应用于数值和位串数据类型。IUT应通过这两种数据类型的测试。

1. 数值算法

本节测试适用于使用整数型或实数数据类型的CHANGE\_OF\_VALUE算法。

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12, 13.3.3和13.9。

目的：验证应用于数值数据类型的CHANGE\_OF\_VALUE事件算法。此测试适用于Event\_Type为CHANGE\_OF\_VALUE的Event Enrollment对象。

配置要求：配置IUT，使Event\_Enable属性值为TRUE（用于TO-NORMAL转换）；Issue\_Confirmed\_Notifications属性值为FALSE；事件生成对象初始态为NORMAL。

测试步骤：除使用UnconfirmedEventNotification服务请求代替ConfirmedEventNotification服务请求发送事件报告和TD不接收消息外，此测试用例的测试步骤与9.4.3.1中的测试步骤相同。

注意事项：除使用UnconfirmedEventNotification服务请求代替ConfirmedEventNotification服务请求发送事件报告外此测试用例的结果与9.4.3.1中的相同。用于这些消息的MAC地址应该是到达TD本地网络的广播，或者是TD的MAC地址。

1. 位串算法

本节测试适用于使用位串数据类型的CHANGE\_OF\_VALUE算法。

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12, 13.3.3和13.9。

目的：验证CHANGE\_OF\_VALUE事件算法。此测试适用于Event\_Type为COV的Event Enrollment对象。

配置要求：配置IUT，使Event\_Enable属性值为TRUE（用于TO-OFFNORMAL和TO-NORMAL转换）；Issue\_Confirmed\_Notifications属性值为FALSE；事件生成对象初始态为NORMAL。

测试步骤：除使用UnconfirmedEventNotification服务请求代替ConfirmedEventNotification服务请求发送事件报告和TD不接收消息外，此测试用例的测试步骤与9.4.3.2中的测试步骤相同。

注意事项：除使用UnconfirmedEventNotification服务请求代替ConfirmedEventNotification服务请求发送事件报告外此测试用例的结果与9.4.3.2中的相同。用于这些消息的MAC地址应该是到达TD本地网络的广播，或者是TD的MAC地址。

1. COMMAND\_FAILURE测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.7，12.12,12.19，13.2，13.3.4和13.9。

目的：验证COMMAND\_FAILURE算法。此测试适用于Event\_Type为COMMAND\_FAILURE的Event Enrollment对象和二进制输出、多状态输出对象的内部事件报告。

配置要求：配置IUT，使Event\_Enable属性值为TRUE（用于TO-OFFNORMAL和TO-NORMAL转换）；Issue\_Confirmed\_Notifications属性值为FALSE；事件生成对象初始态为NORMAL。Feedback\_Value属性应与常用于验证输出的输入信号解耦，以便独立操作。

测试步骤：除使用UnconfirmedEventNotification服务请求代替ConfirmedEventNotification服务请求发送事件报告和TD不接收消息外，此测试用例的测试步骤与9.4.4中的测试步骤相同。

注意事项：除使用UnconfirmedEventNotification服务请求代替ConfirmedEventNotification服务请求发送事件报告外此测试用例的结果与9.4.4中的相同。用于这些消息的MAC地址应该是到达TD本地网络的广播，或者是TD的MAC地址。

1. FLOATING\_LIMIT测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12，12.17，13.2，13.3.5和13.9。

目的：验证FLOATING\_LIMIT事件算法。此测试适用于Event\_Type为FLOATING\_LIMIT的Event Enrollment对象和循环对象的内部事件报告。测试循环对象时，High\_Diff\_Limit和Low\_Diff\_Limit都将被下面的测试描述中的Error\_Limit替换。

配置要求：配置IUT，使Event\_Enable属性值为TRUE（用于TO-OFFNORMAL和TO-NORMAL转换）；Issue\_Confirmed\_Notifications属性值为FALSE；事件生成对象初始态为NORMAL。

测试步骤：除使用UnconfirmedEventNotification服务请求代替ConfirmedEventNotification服务请求发送事件报告和TD不接收消息外，此测试用例的测试步骤与9.4.5中的测试步骤相同。

注意事项：除使用UnconfirmedEventNotification服务请求代替ConfirmedEventNotification服务请求发送事件报告外此测试用例的结果与9.4.5中的相同。用于这些消息的MAC地址应该是到达TD本地网络的广播，或者是TD的MAC地址。

1. OUT\_OF\_RANGE测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.2, 12.3, 12.4, 12.12, 13.2, 13.3.6和13.9。

目的：验证OUT\_OF\_RANGE事件算法。此测试适用于Event\_Type为OUT\_OF\_RANGE的Event Enrollment对象和模拟输入、模拟输出、模拟值对象的内部事件报告。

配置要求：配置IUT，使Event\_Enable属性值为TRUE（用于TO-OFFNORMAL和TO-NORMAL转换）；对使用内部报告的对象的HighLimit和LowLimit事件，Limit\_Enable属性的值应为TRUE；Issue\_Confirmed\_Notifications属性值为FALSE；事件生成对象初始态为NORMAL。

测试步骤：除使用UnconfirmedEventNotification服务请求代替ConfirmedEventNotification服务请求发送事件报告和TD不接收消息外，此测试用例的测试步骤与9.4.6中的测试步骤相同。

注意事项：除使用UnconfirmedEventNotification服务请求代替ConfirmedEventNotification服务请求发送事件报告外此测试用例的结果与9.4.6中的相同。用于这些消息的MAC地址应该是到达TD本地网络的广播，或者是TD的MAC地址。

1. BUFFER\_READY测试

所需测试：ReadProperty服务执行测试，10.18。

BACnet参考条款：12.12, 12.25, 13.2, 13.3.7和13.9。

目的：验证BUFFER\_READY事件算法。此测试适用于支持内部通知的趋势日志对象以及Event\_Type为BUFFER\_READY的Event Enrollment对象。

测试思想、配置要求和测试步骤与9.4.7中的相同，不同之处在于Issue Confirmed Notifications参数的值为FALSE，发出UnconfirmedEventNotification请求而不是ConfirmedEventNotification请求，并且TD不确认接收到通知。

1. CHANGE\_OF\_LIFE\_SAFETY的测试
2. NORMAL到OFFNORMAL转换测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.7, 12.15, 12.16，13.3.8，13.13和图13-9。

目的：验证在NORMAL和OFFNORMAL事件状态之间转换的对象的CHANGE\_OF\_LIFE\_SAFETY算法。此测试适用于Event\_Type为CHANGE\_OF\_LIFE\_SAFETY的Event Enrollment对象以及生命安全点和生命安全区的内部事件报告。

测试方案，配置要求和测试步骤与9.4.8.1相同，除了Issue\_Confirmed\_Notifications参数值为FALSE，使用UnconfirmedEventNotification服务请求而非ConfirmedEventNotification服务请求，且TD不接收报告。消息MAC地址是到达TD的本地网络的广播地址或TD的MAC地址。

1. OFFNORMAL到NORMAL转换测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.7, 12.15, 12.16，13.3.8，13.9和图13-9。

目的：验证在NORMAL和OFFNORMAL事件状态之间转换的对象的CHANGE\_OF\_LIFE\_SAFETY算法。此测试适用于Event\_Type为CHANGE\_OF\_LIFE\_SAFETY的Event Enrollment对象以及生命安全点和生命安全区的内部事件报告。

测试方案，配置要求和测试步骤与9.4.8.2相同，除了Issue\_Confirmed\_Notifications参数值为FALSE，使用UnconfirmedEventNotification服务请求而非ConfirmedEventNotification服务请求，且TD不接收报告。消息MAC地址是到达TD的本地网络的广播地址或TD的MAC地址。

1. NORMAL到LIFE\_SAFETY\_ALARM转换测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.7, 12.15, 12.16，13.3.8，13.9和图13-9。

目的：验证在NORMAL和LIFE\_SAFETY\_ALARM事件状态之间转换的对象的CHANGE\_OF\_LIFE\_SAFETY算法。此测试适用于Event\_Type为CHANGE\_OF\_LIFE\_SAFETY的Event Enrollment对象以及生命安全点和生命安全区的内部事件报告。

测试方案，配置要求和测试步骤与9.4.8.3相同，除了Issue\_Confirmed\_Notifications参数值为FALSE，使用UnconfirmedEventNotification服务请求而非ConfirmedEventNotification服务请求，且TD不接收报告。

1. LIFE\_SAFETY\_ALARM到NORMAL转换测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.7, 12.15, 12.16，13.3.8，13.9和图13-9。

目的：验证在LIFE\_SAFETY\_ALARM和NORMAL事件状态之间转换的对象的CHANGE\_OF\_LIFE\_SAFETY算法。此测试适用于Event\_Type为CHANGE\_OF\_LIFE\_SAFETY的Event Enrollment对象以及生命安全点和生命安全区的内部事件报告。

测试方案，配置要求和测试步骤与9.4.8.4相同，除了Issue\_Confirmed\_Notifications参数值为FALSE，使用UnconfirmedEventNotification服务请求而非ConfirmedEventNotification服务请求，且TD不接收报告。

1. LIFE\_SAFETY\_ALARM到OFFNORMAL转换测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.7, 12.15, 12.16，13.3.8，13.9和图13-9。

目的：验证在LIFE\_SAFETY\_ALARM和OFFNORMAL事件状态之间转换的对象的CHANGE\_OF\_LIFE\_SAFETY算法。此测试适用于Event\_Type为CHANGE\_OF\_LIFE\_SAFETY的Event Enrollment对象以及生命安全点和生命安全区的内部事件报告。

测试方案，配置要求和测试步骤与9.4.8.5相同，除了Issue\_Confirmed\_Notifications参数值为FALSE，使用UnconfirmedEventNotification服务请求而非ConfirmedEventNotification服务请求，且TD不接收报告。

1. OFFNORMAL到LIFE\_SAFETY\_ALARM转换测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.7, 12.15, 12.16，13.3.8，13.9和图13-9。

目的：验证在OFFNORMAL和LIFE\_SAFETY\_ALARM事件状态之间转换的对象的CHANGE\_OF\_LIFE\_SAFETY算法。此测试适用于Event\_Type为CHANGE\_OF\_LIFE\_SAFETY的Event Enrollment对象以及生命安全点和生命安全区的内部事件报告。

测试方案，配置要求和测试步骤与9.4.8.6相同，除了Issue\_Confirmed\_Notifications参数值为FALSE，使用UnconfirmedEventNotification服务请求而非ConfirmedEventNotification服务请求，且TD不接收报告。

1. 事件状态保持时模式转换测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.7, 12.15, 12.16，13.3.8，13.9和图13-9。

目的：验证模式改变时在NORMAL，OFFNORMAL和LIFE\_SAFETY\_ALARM事件状态之间转换的对象的CHANGE\_OF\_LIFE\_SAFETY算法。当模式更改但事件状态不更改时执行测试，当模式改变与事件状态改变同时发生时也进行测试。在后一种情况下，测试验证通知立即进行而不等待时间延迟。测试适用于Event\_Type为CHANGE\_OF\_LIFE\_SAFETY的Event Enrollment对象以及生命安全点和生命安全区对象的内部事件报告。

测试方案，配置要求和测试步骤与9.4.8.7相同，除了Issue\_Confirmed\_Notifications参数值为FALSE，使用UnconfirmedEventNotification服务请求而非ConfirmedEventNotification服务请求，且TD不接收报告。

1. NORMAL到OFFNORMAL模式转换测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.7, 12.15, 12.16，13.3.8，13.9和图13-9。

目的：验证模式改变时在NORMAL，OFFNORMAL事件状态之间转换的对象的CHANGE\_OF\_LIFE\_SAFETY算法。测试适用于Event\_Type为CHANGE\_OF\_LIFE\_SAFETY的Event Enrollment对象以及生命安全点和生命安全区对象的内部事件报告。

测试方案，配置要求和测试步骤与9.4.8.8相同，除了Issue\_Confirmed\_Notifications参数值为FALSE，使用UnconfirmedEventNotification服务请求而非ConfirmedEventNotification服务请求，且TD不接收报告。

1. OFFNORMAL到NORMAL模式转换测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.7, 12.15, 12.16，13.3.8，13.9和图13-9。

目的：验证模式改变时在OFFNORMAL，NORMAL事件状态之间转换的对象的CHANGE\_OF\_LIFE\_SAFETY算法。测试适用于Event\_Type为CHANGE\_OF\_LIFE\_SAFETY的Event Enrollment对象以及生命安全点和生命安全区对象的内部事件报告。

测试方案，配置要求和测试步骤与9.4.8.9相同，除了Issue\_Confirmed\_Notifications参数值为FALSE，使用UnconfirmedEventNotification服务请求而非ConfirmedEventNotification服务请求，且TD不接收报告。

1. NORMAL到LIFE\_SAFETY\_ALARM模式转换测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.7, 12.15, 12.16，13.3.8，13.9和图13-9。

目的：验证模式改变时在NORMAL和LIFE\_SAFETY\_ALARM事件状态之间转换的对象的CHANGE\_OF\_LIFE\_SAFETY算法。测试适用于Event\_Type为CHANGE\_OF\_LIFE\_SAFETY的Event Enrollment对象以及生命安全点和生命安全区对象的内部事件报告。

测试方案，配置要求和测试步骤与9.4.8.10相同，除了Issue\_Confirmed\_Notifications参数值为FALSE，使用UnconfirmedEventNotification服务请求而非ConfirmedEventNotification服务请求，且TD不接收报告。

1. LIFE\_SAFETY\_ALARM 到NORMAL模式转换测验

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.7, 12.15, 12.16，13.3.8，13.9和图13-9。

目的：验证模式改变时在LIFE\_SAFETY\_ALARM和NORMAL事件状态之间转换的对象的CHANGE\_OF\_LIFE\_SAFETY算法。测试适用于Event\_Type为CHANGE\_OF\_LIFE\_SAFETY的Event Enrollment对象以及生命安全点和生命安全区对象的内部事件报告。

测试方案，配置要求和测试步骤与9.4.8.11相同，除了Issue\_Confirmed\_Notifications参数值为FALSE，使用UnconfirmedEventNotification服务请求而非ConfirmedEventNotification服务请求，且TD不接收报告。

1. LIFE\_SAFETY\_ALARM 到OFFNORMAL模式转换测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.7, 12.15, 12.16，13.3.8，13.9和图13-9。

目的：验证模式改变时在LIFE\_SAFETY\_ALARM和OFFNORMAL事件状态之间转换的对象的CHANGE\_OF\_LIFE\_SAFETY算法。测试适用于Event\_Type为CHANGE\_OF\_LIFE\_SAFETY的Event Enrollment对象以及生命安全点和生命安全区对象的内部事件报告。

测试方案，配置要求和测试步骤与9.4.8.12相同，除了Issue\_Confirmed\_Notifications参数值为FALSE，使用UnconfirmedEventNotification服务请求而非ConfirmedEventNotification服务请求，且TD不接收报告。

1. OFFNORMAL到LIFE\_SAFETY\_ALARM模式转换测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.7, 12.15, 12.16，13.3.8，13.9和图13-9。

目的：验证模式改变时在OFFNORMAL和LIFE\_SAFETY\_ALARM事件状态之间转换的对象的CHANGE\_OF\_LIFE\_SAFETY算法。测试适用于Event\_Type为CHANGE\_OF\_LIFE\_SAFETY的Event Enrollment对象以及生命安全点和生命安全区对象的内部事件报告。

测试方案，配置要求和测试步骤与9.4.8.13相同，除了Issue\_Confirmed\_Notifications参数值为FALSE，使用UnconfirmedEventNotification服务请求而非ConfirmedEventNotification服务请求，且TD不接收报告。

1. TO-FAULT转换测试

所需测试：ReadProperty服务执行测试，10.18；WriteProperty服务执行测试，10.22。

BACnet参考条款：12.12.7, 12.15, 12.16，13.3.8，13.9和图13-9。

目的：验证模式改变时在NORMAL和FAULT事件状态之间转换的对象的CHANGE\_OF\_LIFE\_SAFETY算法。测试适用于生命安全点和生命安全区对象的内部事件报告。

测试方案，配置要求和测试步骤与9.4.8.14相同，除了Issue\_Confirmed\_Notifications参数值为FALSE，使用UnconfirmedEventNotification服务请求而非ConfirmedEventNotification服务请求，且TD不接收报告。

* 1. 获取警报摘要（GetAlarmSummary）服务启动测试

目的：验证IUT可启动GetAlarmSummary服务请求。

所需测试：无。

BACnet参考条款：13.10。

测试步骤：

1. RECEIVE GetAlarmSummary-Request

2. TRANSMIT BACnet-ComplexACK-PDU,

‘List of Alarm Summaries’ = (an empty list)

* 1. 获取事件摘要（GetEnrollmentSummary）服务启动测试

该节定义了支持启动GetEnrollmentSummary服务请求的必要测试。

所需测试：无。

BACnet参考条款：13.11。

1. 确认过滤器

目的：验证IUT可启动传送确认过滤器的GetEnrollmentSummary服务请求。

测试步骤：

1. RECEIVE GetEnrollmentSummary-Request,

‘Acknowledgment Filter’ = (any valid Acknowledgment Filter enumeration)

2. TRANSMIT BACnet-ComplexACK-PDU,

‘List of Alarm Summaries’ = (an empty list)

1. 注册过滤器

目的：验证IUT可启动传送注册过滤器的GetEnrollmentSummary服务请求。

测试步骤：

1. RECEIVE GetEnrollmentSummary-Request,

‘Enrollment Filter’ = (any valid BACnetRecipientProcess)

2. TRANSMIT BACnet-ComplexACK-PDU,

‘List of Alarm Summaries’ = (an empty list)

1. 事件状态过滤器

目的：验证IUT可启动传送事件状态过滤器的GetEnrollmentSummary服务请求。

测试步骤：

1. RECEIVE GetEnrollmentSummary-Request,

‘Event State Filter’ = (any valid Event State Filter enumeration)

2. TRANSMIT BACnet-ComplexACK-PDU,

‘List of Alarm Summaries’ = (an empty list)

1. 事件类型过滤器

目的：验证IUT可启动传送事件类型过滤器的GetEnrollmentSummary服务请求。

测试步骤：

1. RECEIVE GetEnrollmentSummary-Request,

‘Event Type Filter’ = (any valid BACnetEventType enumeration)

2. TRANSMIT BACnet-ComplexACK-PDU,

‘List of Alarm Summaries’ = (an empty list)

1. 优先过滤器

目的：验证IUT可启动传送优先过滤器的GetEnrollmentSummary服务请求。

测试步骤：

1. RECEIVE GetEnrollmentSummary-Request,

‘Priority Filter’ = (any valid priority range)

2. TRANSMIT BACnet-ComplexACK-PDU,

‘List of Alarm Summaries’ = (an empty list)

1. 通知类别过滤器

目的：验证IUT可启动传送通知类别过滤器的GetEnrollmentSummary服务请求。

测试步骤：

1. RECEIVE GetEnrollmentSummary-Request,

‘Notification Class Filter’ = (any valid Notification Class)

2. TRANSMIT BACnet-ComplexACK-PDU,

‘List of Alarm Summaries’ = (an empty list)

1. 多个过滤器

目的：验证IUT可启动传送多个Filter的GetEnrollmentSummary服务请求。

测试步骤：

1. RECEIVE GetEnrollmentSummary-Request,

‘Acknowledgment Filter’ = (any valid Acknowledgment Filter enumeration)

‘Enrollment Filter’ = (any valid BACnetRecipientProcess)

‘Event State Filter’ = (any valid Event State Filter enumeration)

‘Event Type Filter’ = (any valid BACnetEventType enumeration)

‘Priority Filter’ = (any valid priority range)

‘Notification Class Filter’ = (any valid Notification Class)

2. TRANSMIT BACnet-ComplexACK-PDU,

‘List of Alarm Summaries’ = (an empty list)

* 1. 获取事件信息（GetEventInformation）服务启动测试

该节定义了证明IUT支持发起GetEventInformation服务请求的必要测试。

所需测试：无。

BACnet参考条款：13.12。

1. 无链接

目的：验证IUT可启动简单的GetEventInformation服务请求。

测试步骤：

1. RECEIVE GetEventInformation-Request,

‘Last Received Object Identifier’ = (none)

2. TRANSMIT BACnet-ComplexACK-PDU,

‘List of Event Summaries’ = (any valid list)

‘More Events’ = FALSE

3. CHECK (that the IUT exhibits the vendor defined results)

1. 有链接

目的：验证IUT可以通过启动具有链接参数“最后接收的对象标识符”的GetEventInformation服务请求来响应“更多事件”标志。

测试步骤：

1. RECEIVE GetEventInformation-Request,

‘Last Received Object Identifier’ = (none)

2. TRANSMIT BACnet-ComplexACK-PDU,

‘List of Event Summaries’ = (a non-empty list)

‘More Events’ = TRUE

3. BEFORE the tester’s patience is exceeded

RECEIVE GetEventInformation-Request,

‘Last Received Object Identifier’ = (last object identifier sent in step 2)

4. TRANSMIT BACnet-ComplexACK-PDU,

‘List of Event Summaries’ = (a non-empty list)

‘More Events’ = FALSE

5. CHECK (that the IUT exhibits the vendor defined results)

* 1. 生命安全操作（LifeSafetyOperation）服务启动测试

1. 一个对象的LifeSafetyOperation服务启动测试

该节定义了证明IUT可启动单个生命安全点或生命安全区对象的LifeSafetyOperation服务请求的测试。

所需测试：无。

BACnet参考条款：13.13。

目的：证明IUT可正确发起一个LifeSafetyOperation服务请求。

测试步骤：

1. RECEIVE LifeSafetyOperation-Request,

‘Requesting Process Identifier’ = (any valid identifier),

‘Requesting Source’ = (any valid character string),

‘Request’ = (any valid LifeSafetyOperation request),

‘Object Identifier’ = (any valid Life Safety Point or Life Safety Zone object identifier)

1. 设备所有对象的LifeSafetyOperation服务启动测试

该节定义了证明IUT可启动设备所有生命安全点和生命安全区域对象的LifeSafetyOperation服务请求的测试。

所需测试：无。

BACnet参考条款：13.13。

目的：证明IUT可正确发起一个对生命安全对象的LifeSafetyOperation服务请求。

测试步骤：

1. RECEIVE LifeSafetyOperation-Request,

‘Requesting Process Identifier’ = (any valid identifier),

‘Requesting Source’ = (any valid character string),

‘Request’ = (any valid LifeSafetyOperation request)

* 1. 订阅COV（SubscribeCOV）服务启动测试

该节定义了执行发起 SubscribeCOV服务请求的必要测试。

所需测试：无。

BACnet参考条款：13.14。

1. 确认通知订阅

目的：证明IUT可对确认的通知启动一个SubscribeCOV服务请求。

测试步骤：

1. RECEIVE SubscribeCOV-Request,

‘Subscriber Process Identifier’ = (any valid process identifier),

‘Monitored Object Identifier’ = (any identifier for a standard object type for which COV reporting is defined),

‘Issue Confirmed Notifications’ = TRUE,

‘Lifetime’ = (any non-zero value)

2. TRANSMIT BACnet-SimpleACK-PDU

1. 未确认通知订阅

目的：证明IUT可对未确认通知启动一个SubscribeCOV服务请求。

测试步骤：

1. RECEIVE SubscribeCOV-Request,

‘Subscriber Process Identifier’ = (any valid process identifier),

‘Monitored Object Identifier’ = (any identifier for a standard object type for which COV reporting is defined),

‘Issue Confirmed Notifications’ = FALSE,

‘Lifetime’ = (any non-zero value)

2. TRANSMIT BACnet-SimpleACK-PDU

1. 取消订阅

目的：证明IUT可发起SubscribeCOV服务请求来取消订阅。

测试步骤：

1. RECEIVE SubscribeCOV-Request,

‘Subscriber Process Identifier’ = (any valid process identifier),

‘Monitored Object Identifier’ = (any identifier for a standard object type for which COV reporting is defined)

2. TRANSMIT BACnet-SimpleACK-PDU

* 1. 订阅COV属性（SubscribeCOVProperty）服务启动测试

该节定义了证明IUT可发起SubscribeCOV属性服务请求的测试。

所需测试：无。

BACnet参考条款：13.15。

1. 确认通知订阅

目的：验证IUT是否可以发起SubscribeCOV属性服务请求来确认通知。

测试步骤：

1. RECEIVE SubscribeCOVProperty-Request,

‘Subscriber Process Identifier’ = (any valid process identifier),

‘Monitored Object Identifier’ = (any valid object identifier),

‘Issue Confirmed Notifications’ = TRUE,

‘Lifetime’ = (any non-zero value),

‘Monitored Property Identifier’ = (any valid property identifier),

‘COV Increment’ = (any valid value – optional)

2. TRANSMIT BACnet-SimpleACK-PDU

1. 未确认通知订阅

目的：验证IUT是否可以发起SubscribeCOV属性服务请求来否认通知。

测试步骤：

1. RECEIVE SubscribeCOVProperty-Request,

‘Subscriber Process Identifier’ = (any valid process identifier),

‘Monitored Object Identifier’ = (any valid object identifier),

‘Issue Confirmed Notifications’ = FALSE,

‘Lifetime’ = (any non-zero value),

‘Monitored Property Identifier’ = (any valid property identifier),

‘COV Increment’ = (any valid value – optional)

2. TRANSMIT BACnet-SimpleACK-PDU

1. 取消订阅

目的：验证IUT是否启动SubscribeCOV属性服务请求来取消订阅。

测试步骤：

1. RECEIVE SubscribeCOVProperty-Request,

‘Subscriber Process Identifier’ = (any valid process identifier),

‘Monitored Object Identifier’ = (any valid object identifier),

‘Monitored Property Identifier’ = (any valid property identifier),

‘COV Increment’ = (any valid value – optional)

2. TRANSMIT BACnet-SimpleACK-PDU

* 1. 原子读取文件（AtomicReadFile）启动测试

该节定义了证明可发起AtomicReadFile服务请求的必要测试。BACnet文件访问服务允许以记录为基础或以流为基础访问文件。如果IUT支持基于流的文件结构，则应执行流访问测试（9.12.1）。如果IUT支持基于记录的文件结构，则应执行记录访问测试（9.12.2）。

所需测试：无。

BACnet参考条款：14.1。

1. 流访问

目的：证明IUT可使用基于流的文件访问方法发起AtomicReadFile服务请求。

测试步骤：

1. RECEIVE AtomicReadFile-Request,

‘File Start Position’ = (any value ≥ 0),

‘Requested Octet Count’ = (any value > 0)

2. TRANSMIT AtomicReadFile-ACK,

‘End Of File’ = TRUE,

‘File Start Position’ = (the start position indicated in step 1),

‘File Data’ = (any stream file data of size ≤ ‘Requested Octet Count’ from step 1)

3. CHECK (if the IUT displays the file data, verify that it is correct)

1. 记录访问

目的：证明IUT可使用基于记录的文件访问方法发起AtomicReadFile服务请求。

测试步骤：

1. RECEIVE AtomicReadFile-Request,

‘File Start Record’ = (any value ≥ 0),

‘Requested Record Count’ = (any value > 0)

2. TRANSMIT AtomicReadFile-ACK,

‘End Of File’ = TRUE,

‘File Start Record’ = (the start position indicated in step 1),

‘Returned Record Count’ = (any value ≤ the ‘Requested Record Count’ from step 1),

‘File Record Data’ = (any record file data containing the indicated number of records)

3. CHECK (if the IUT displays the file data, verify that it is correct)

* 1. 原子写文件（AtomicWriteFile）服务启动测试

该节定义了证明可发起AtomicWriteFile服务请求的必要测试。BACnet文件访问服务允许以记录为基础或以流为基础访问文件。如果IUT支持基于流的文件结构，则应执行流访问测试（9.13.1）。如果IUT支持基于记录的文件结构，则应执行记录访问测试（9.13.2）。

所需测试：无。

BACnet参考条款：14.2。

1. 流访问

目的：证明IUT可使用基于流的文件访问方法发起AtomicWriteFile服务请求。

测试步骤：

1. RECEIVE AtomicWriteFile-Request,

‘File Start Position’ = (any position ≥ -1),

‘File Data’ = (any stream file data containing at least one octet)

2. TRANSMIT AtomicWriteFile-ACK,

‘File Start Position’ = (0 if the ‘File Start Position was -1, otherwise the indicated start position)

1. 记录访问

目的：证明IUT可使用基于记录的文件访问方法发起AtomicWriteFile服务请求。

测试步骤：

1. RECEIVE AtomicWriteFile-Request,

‘File Start Record’ = (any record ≥ -1),

‘Record Count’ = (any value > 0),

‘File Record Data’ = (any record file data containing the indicated number of records)

2. TRANSMIT AtomicWriteFile-ACK,

‘File Start Record’ = (0 if the ‘File Start Position was -1, otherwise the indicated start position)

* 1. 添加列表元素（AddListElement）服务启动测试

该节定义了证明可发起AddListElement服务请求的测试。

所需测试：无。

BACnet参考条款：15.1。

1. 非数组属性

目的：验证IUT是否可以启动不包含“Property Array Index”参数的AddListElement服务请求。

测试步骤：

1. RECEIVE AddListElement-Request,

‘Object Identifier’ = (any object that contains a property having a list datatype),

‘Property Identifier’ = (any property having a list datatype),

‘List of Elements’ = (two or more elements with the correct datatype to add to the list)

2. TRANSMIT BACnet-SimpleACK-PDU

1. 数组属性

目的：验证IUT是否可以启动包含“Property Array Index”参数的AddListElement服务请求。

测试步骤：

1. RECEIVE AddListElement-Request,

‘Object Identifier’ = (any object that contains a property having a datatype that is an array of lists),

‘Property Identifier’ = (any property having a datatype that is an array of lists),

‘Property Array Index’ = (any value > 0),

‘List of Elements’ = (two or more elements with the correct datatype to add to the list)

2. TRANSMIT BACnet-SimpleACK-PDU

* 1. 移除列表元素（RemoveListElement）服务启动测试

该节定义了证明可发起RemoveListElement服务请求的必要测试。

所需测试：无。

BACnet参考条款：15.2。

1. 非数组属性

目的：验证IUT是否可以启动不包含“Property Array Index”参数的RemoveListElement服务请求。

测试步骤：

1. RECEIVE RemoveListElement-Request,

‘Object Identifier’ = (any object that contains a property having a list datatype),

‘Property Identifier’ = (any property having a list datatype),

‘List of Elements’ = (two or more elements with the correct datatype to removefrom the list)

2. TRANSMIT BACnet-SimpleACK-PDU

1. 数组属性

目的：验证IUT是否可以启动不包含“Property Array Index”参数的RemoveListElement服务请求。

测试步骤：

1. RECEIVE RemoveListElement-Request,

‘Object Identifier’ = (any object that contains a property having a datatype that is an array of lists),

‘Property Identifier’ = (any property having a datatype that is an array of lists),

‘Property Array Index’ = (any value > 0),

List of Elements’ = (two or more elements with the correct datatype to remove from the list)

2. TRANSMIT BACnet-SimpleACK-PDU

* 1. 创建对象（CreateObject）服务启动测试

该节定义了证明可发起CreateObject服务请求的测试。

所需测试：无。

BACnet参考条款：15.3。

1. 通过指定没有初始值的对象标识符创建对象

目的：验证IUT是否可以启动包含Object Specifier参数且没有初始属性值的对象标识符的CreateObject服务请求。

测试步骤：

1. RECEIVE CreateObject-Request,

‘Object Specifier’ = (any BACnetObjectIdentifier)

2. TRANSMIT CreateObject-ACK,

‘Object Identifier’ = (the object identifier specified in step 1)

1. 通过指定没有初始值的对象类型创建对象

目的：验证IUT是否可以启动包含Object Specifier参数且没有初始属性值的对象类型的CreateObject服务请求。

测试步骤：

1. RECEIVE CreateObject-Request,

‘Object Specifier’ = (any BACnetObjectType),

2. TRANSMIT CreateObject-ACK,

‘Object Identifier’ = (an object identifier consistent with the object type specified in step 1)

1. 通过指定对象标识符并提供初始值创建对象

目的：验证IUT是否可以启动包含Object Specifier参数以及要创建的对象的初始属性值列表的CreateObject服务请求。

测试步骤：

1. RECEIVE CreateObject-Request,

‘Object Specifier’ = (any BACnetObjectIdentifier)

‘List of Initial Values’ = (a list of one or more properties and their initial values that the IUT will accept)

2. TRANSMIT CreateObject-ACK,

‘Object Identifier’ = (the object identifier specified in step 1)

1. 通过指定对象类型并提供初始值创建对象

目的：验证IUT是否可以启动包含Object Specifier参数以及要创建的对象的初始属性值对象类型的CreateObject服务请求。

测试步骤：

1. RECEIVE CreateObject-Request,

‘Object Specifier’ = (any BACnetObjectType),

‘List of Initial Values’ = (a list of one or more properties and their initial values that the IUT will accept)

2. TRANSMIT CreateObject-ACK,

‘Object Identifier’ = (an object identifier consistent with the object type specified in step 1)

* 1. 删除对象（DeleteObject）服务启动测试

该节定义了证明IUT可发起DeleteObject服务请求的测试。

所需测试：无。

BACnet参考条款：15.4。

目的：证明IUT可发起DeleteObject服务请求。

测试步骤：

1. RECEIVE DeleteObject-Request,

‘Object Identifier’ = (any object identifier)

2. TRANSMIT BACnet-SimpleACK-PDU

* 1. 读属性（ReadProperty）服务启动测试

该节定义了必要的测试，证明可发起ReadProperty服务请求的测试。

所需测试：无。

BACnet参考条款：15.5。

1. 读取非数组属性

目的：验证IUT是否可以启动不包含“Property Array Index”参数的ReadProperty服务请求，并可正确处理响应。

测试步骤：

1. RECEIVE ReadProperty-Request,

‘Object Identifier’ = (any object),

‘Property Identifier’ = (any valid non-array property of the specified object)

2. TRANSMIT BACnet-ComplexACK-PDU,

‘Object Identifier’ = (object identifier from step 1),

‘Property Identifier’ = (property identifier from step 1),

‘Property Value’ = (any valid value for the property)

3. CHECK (that the IUT exhibits the vendor defined results)

1. 读取数组元素

目的：验证IUT是否可以启动引用数组属性的特定元素的ReadProperty服务请求，并可正确处理响应。

测试步骤：

1. RECEIVE ReadProperty-Request,

‘Object Identifier’ = (any object),

‘Property Identifier’ = (any valid array property of the specified object),

‘Array Index’ = (any valid array index for the specified property)

2. TRANSMIT BACnet-ComplexACK-PDU,

‘Object Identifier’ = (object identifier from step 1),

‘Property Identifier’ = (property identifier from step 1),

Array Index’ = (array index from step 1),

‘Property Value’ = (any valid value for the property)

3. CHECK (that the IUT exhibits the vendor defined results)

1. 读取和显示属性

目的：验证IUT是否能够读取和显示属性，此测试是用于测试数据表示要求的通用测试。

配置：对于此测试，选择对象O1中的属性P1。TD应配置为不支持执行ReadPropertyMultiple或启动COV报告。

测试步骤：

1. MAKE (the IUT read P1)

2. RECEIVE ReadProperty-Request,

‘Object Identifier’ = O1,

‘Property Identifier’ = P1

3. TRANSMIT BACnet-ComplexACK-PDU,

‘Object Identifier’ = O1,

‘Property Identifier’ = P1,

‘Property Value’ = (any valid value for P1)

4. CHECK (that the IUT presents a value that is consistent with the value received in step 3)

注意事项：由于舍入、截断、格式化、语言转换等原因，IUT提供的值可能与传输值不同。

如果正在读取和显示的属性是一个数组，则在步骤2中IUTReadProperty请求可能包含一个数组索引参数，步骤4中TD的响应可能包含数组索引参数。

如果IUT不确定TD是否支持执行ReadPropertyMultiple、SubscribeCOV和SubscribeCOVProperty，则IUT可支持这些服务。这种情况下，只有当IUT从TD收到正确的BACnetReject-PDU(s)自动回退到使用ReadProperty时才能通过测试，表明不支持其他服务。

1. 读取整个数组的属性

目的：验证IUT可以发起不包含数组属性的Property Array Index参数的ReadProperty服务请求，并且可以正确处理响应。

测试步骤：

1. RECEIVE ReadProperty-Request,

'Object Identifier' = (any object),

'Property Identifier' = (any valid array property of the specified object)

2. TRANSMIT BACnet-ComplexACK-PDU,

'Object Identifier' = (object identifier from step 1),

'Property Identifier' = (property identifier from step 1),

'Property Value' = (any valid array of values for the property)

3. CHECK (that the IUT exhibits the vendor defined results)

1. 读取数组大小

目的：验证IUT可以针对数组属性的大小启动ReadProperty服务请求，并可以正确处理响应。

测试步骤：

1. RECEIVE ReadProperty-Request,

'Object Identifier' = (any object),

'Property Identifier' = (any valid array property of the specified object),

'Array Index' = 0

2. TRANSMIT BACnet-ComplexACK-PDU,

'Object Identifier' = (object identifier from step 1),

'Property Identifier' = (property identifier from step 1),

Array Index' = 0,

'Property Value' = (any valid size for the array)

3. CHECK (that the IUT exhibits the vendor defined results)

* 1. 读取条件属性（ReadPropertyConditional）服务启动测试

本节定义了证明执行支持ReadPropertyConditional服务请求的必要的测试。

所需测试：无。

BACnet参考条款：15.6。

1. 读取符合选择标准的对象的对象标识符

目的：验证IUT可以发起ReadPropertyConditional服务请求，其中请求根据某些选择标准来选择对象，并且没有指定“List of Property References”。

测试步骤：

1. RECEIVE ReadPropertyConditional-Request,

'Selection Logic' = AND,

'List of Selection Criteria' = ((any non-array property identifier, any relation specifier, any valid comparison value),

(any array property identifier, any array index, any relation specifier, any valid comparison value)

)

1. 读取符合选择标准的对象的特定属性

目的：验证IUT可以发起ReadPropertyConditional服务请求，其中请求根据某些选择标准来选择对象，并指定一组要读取的属性。

测试步骤：

1. RECEIVE ReadPropertyConditional-Request,

'Selection Logic' = OR,

'List of Selection Criteria' = ((any non-array property identifier, any relation specifier, any valid comparison value),

(any array property identifier, any array index, any relation specifier, any valid comparison value)

),

'List of Property References' = (two or more properties that correspond to objects meeting the selection criteria)

* 1. 读取多个属性（ReadPropertyMultiple）服务启动测试

目的：该子句定义了证明对启动ReadPropertyMultiple服务请求支持的必要的测试。为了说明启动ReadPropertyMultiple服务所需的功能，至少需要支持9.20.2到9.20.4中定义的至少一个组合。

所需测试：无。

BACnet参考条款：15.7。

1. 读一个对象的一个属性

目的：验证IUT可以正确地发起一个包含单个对象的单个属性的ReadPropertyMultiple服务请求。如果不能将IUT配置为发起一个包含单个对象的单个属性的ReadPropertyMultiple服务请求，则省略此测试。

测试步骤：

1. RECEIVE ReadPropertyMultiple-Request,

'Object Identifier' = (any valid object identifier),

'List of Property References' = (a single property of the specified object)

1. 读一个对象的多个属性

目的：验证IUT可以正确地发起一个包含单个对象的多个属性的ReadPropertyMultiple服务请求。如果不能将IUT配置为发起包含单个对象的多个属性的ReadPropertyMultiple服务请求，则省略此测试。

测试步骤：

1. RECEIVE ReadPropertyMultiple-Request,

'Object Identifier' = (any valid object identifier),

'List of Property References' = (two or more properties of the specified object)

1. 读取多个对象，每个对象一个属性

目的：验证IUT是否可以正确启动一个ReadPropertyMultiple服务请求，该请求包含多个对象和每个对象的单个属性。如果不能将IUT配置为发起一个ReadPropertyMultiple服务请求，该服务请求包含多个对象和每个对象的单个属性，则省略此测试。

测试步骤：

1. RECEIVE ReadPropertyMultiple-Request,

'Object Identifier' = (any valid object identifier),

'List of Property References' = (a single property of the specified object),

'Object Identifier' = (any valid object identifier not previously used),

'List of Property References' = (a single property of the specified object),

-- … (Including additional objects and properties is acceptable.)

1. 读取多个对象，每个对象有多个属性

目的：验证IUT可以正确启动一个ReadPropertyMultiple服务请求，该请求包含多个对象以及每个对象的多个属性。如果无法将IUT配置为启动包含多个对象和每个对象多个属性的ReadPropertyMultiple服务请求，则省略此测试。

测试步骤：

1. RECEIVE ReadPropertyMultiple-Request,

'Object Identifier' = (any valid object identifier),

'List of Property References' = (two or more properties of the specified object)

'Object Identifier' = (any valid object identifier not previously used),

'List of Property References' = (two or more properties of the specified object)

-- … (Including additional objects and properties is acceptable.)

1. ReadPropertyMultiple失败后，使用ReadProperty

本节中定义的测试用于验证与不支持ReadPropertyMultiple服务的设备进行互操作时，引发ReadPropertyMultiple的IUT可通过ReadProperty服务获取外部属性值。

1. IUT确定TD不支持ReadPropertyMultiple服务

目的：当IUT确定TD不支持ReadPropertyMultiple服务时，验证IUT能够自动将其服务选择从ReadPropertyMultiple更改为ReadProperty。

测试思想：配置IUT使它通过ReadPropertyMultiple服务访问TD中的一个或多个属性。但在发送ReadPropertyMultiple请求之前，IUT确定TD不支持ReadPropertyMultiple服务，IUT通过ReadProperty服务访问TD的属性值（假定IUT将通过读取TD的Protocol\_Services\_Supported属性来进行此确定，但是该测试明确地不尝试验证此方式）。

配置要求：配置TD使其不支持ReadPropertyMultiple服务。配置IUT，使其能够通过ReadProperty和ReadPropertyMultiple服务访问TD中单个对象的一个或多个属性。如果无法以这种方式配置IUT，则应省略此测试。

测试步骤：

1. MAKE (a condition in the IUT that would normally cause it to send a ReadPropertyMultiple request to the TD to access one or more properties values of a single object)

2. WAIT (a time interval specified by the vendor as sufficient for the IUT to determine that the TD does not support the ReadPropertyMultiple service)

3. REPEAT X = (the properties that the IUT is to read) DO {

RECEIVE ReadProperty-Request,

'Object Identifier' = (object identifier referenced by X),

'Property Identifier' = (property identifier referenced by X)

TRANSMIT ReadProperty-Ack,

'Object Identifier' = (object identifier referenced by X),

'Property Identifier' = (property identifier referenced by X),

'Property Value' = (any valid value)

}

1. Reject - UNRECOGNIZED\_SERVICE应答时，回退到ReadProperty

BACnet参考条款：15.5和15.7

目的：验证当TD返回Reject-PDU和UNRECOGNIZED\_SERVICE的拒绝原因时，IUT能够自动将其服务选择从ReadPropertyMultiple更改为ReadProperty。

测试思想：IUT配置为向TD发送ReadPropertyMultiple请求，以访问单个对象的一个或多个属性。 TD使用Reject-PDU和UNRECOGNIZED\_SERVICE的拒绝原因进行响应。如果没有其他配置，则IUT将一个或多个ReadProperty请求发送到TD，其中每个ReadProperty请求都从原始ReadPropertyMultiple请求中指定一个单独的属性。

配置要求：配置TD使其不支持ReadPropertyMultiple服务。配置IUT使其尝试使用ReadPropertyMultiple服务从TD获取值，而无需先询问TD的Protocol\_Services\_Supported属性。如果无法以这种方式配置IUT，则应省略此测试。

测试步骤：

1. RECEIVE ReadPropertyMultiple-Request,

'Object Identifier' = (object identifier of the specified object),

'List of Property References' = (one or more properties of the specified object)

2. TRANSMIT BACnet-Reject-PDU,

'Reject Reason' = UNRECOGNIZED\_SERVICE

3. REPEAT X = (the properties from Step 1) DO {

RECEIVE ReadProperty-Request,

'Object Identifier' = (object identifier referenced by X),

'Property Identifier' = (property identifier referenced by X)

TRANSMIT ReadProperty-Ack,

'Object Identifier' = (object identifier referenced by X),

'Property Identifier' = (property identifier referenced by X),

'Property Value' = (any valid value)

}

* 1. 读范围（ReadRange）服务启动测试

本节定义了证明对启动ReadRange服务请求的支持的必要测试。

所需测试：无。

BACnet参考条款：15.8。

1. 读取没有指定范围的值

目的：验证IUT可以正确启动一个未指定要返回的任何值范围的ReadRange服务请求。

测试步骤：

1. RECEIVE ReadRange-Request,

'Object Identifier' = (any Trend Log object),

'Property Identifier' = Log\_Buffer

1. 使用数组索引读取值

目的：验证IUT可以正确地启动指定数组索引的ReadRange服务请求。

测试步骤：

1. RECEIVE ReadRange-Request,

'Object Identifier' = (any object with a property that is an array of lists),

'Property Identifier' = (any property that is an array of lists),

'Array Index' = (an value > 0)

1. 按位置读取值范围

目的：为了验证IUT可以正确启动指定按位置返回的值的范围的ReadRange服务请求。

测试步骤：

1. RECEIVE ReadRange-Request,

'Object Identifier' = (any Trend Log object),

'Property Identifier' = Log\_Buffer,

'Reference Index' = (any Unsigned value),

'Count' = (any INTEGER value)

1. 按时间读取一系列值

目的：验证IUT可以正确启动指定按时间返回的值的范围的ReadRange服务请求。

测试步骤：

1. RECEIVE ReadRange-Request,

'Object Identifier' = (any Trend Log object),

'Property Identifier' = Log\_Buffer,

'Reference Time' = (any BACnetDateTime value),

'Count' = (any INTEGER value)

1. 按时间范围读取值范围

目的：验证IUT是否可以正确启动指定了按时间范围返回的值的范围的ReadRange服务请求。

测试步骤：

1. RECEIVE ReadRange-Request,

'Object Identifier' = (any Trend Log object),

'Property Identifier' = Log\_Buffer,

'Beginning Time' = (any BACnetDateTime value),

'Ending Time' = (any BACnetDateTime value)

1. 使用任何有效范围读取项目范围响应BUFFER\_READY事件类型的ConfirmedEventNotification

目的：通过响应BUFFER\_READY事件类型的ConfirmedEventNotification消息，验证IUT是否可以发起一系列ReadRange服务请求来访问连续的日志记录集。

测试思想：TD包含一个日志对象，该对象具有两个逻辑记录集S1和S2。日志继续累积日志记录，直到将逻辑记录集S3和S4添加到缓冲区中为止。TD将有关S2和S3添加的BUFFER\_READY事件通知发送到IUT。IUT通过使用任何有效范围发送一系列ReadRange请求来访问S2和S3中的记录。然后，测试将验证IUT是否访问了S2，S3和S4的某些部分中的一组连续记录。集合S1，S2，S3和S4的大小应足够小，以至于没有一个导致参考服务器发送事件通知。

该测试验证IUT不会每次都简单地读取整个缓冲区，也不会丢失Log\_Buffer中的样本。

配置要求：TD包含一个日志对象L1，该对象具有足够数量的Buffer\_Size属性，它可以包含记录集S1、S2、S3和S4，其中每个记录集的大小足够大，以致TD无法通过单个ReadRange响应返回整个集合（此行为可以通过禁用TD传输分段消息的能力或限制TD可以传输的段数来实现）。配置TD为在测试期间向其Log\_Buffer添加条目，并且在添加S2和S3之后将向IUT发送BUFFER\_READY类型的ConfirmedEventNotification。

测试步骤：

1. MAKE (L1 contain record sets S1 + S2)

2. TRANSMIT ConfirmedEventNotification-Request,

'Subscriber Process Identifier' = (any valid value selected for the IUT),

'Initiating Device Identifier' = TD,

'Event Object Identifier' = (the object detecting the event),

'Time Stamp' = (any valid value),

'Notification Class' = (any valid notification class),

'Priority' = (any valid value),

'Event Type' = BUFFER\_READY,

'Notify Type' = ALARM | EVENT,

'AckRequired' = TRUE | FALSE,

'From State' = NORMAL,

'To State' = NORMAL,

'Event Values' = (L1, Log\_Buffer),

(any Unsigned32 value that indicates the end of S1),

(any Unsigned32 value that indicates the addition of S2)

3. RECEIVE Simple-Ack-PDU

4. BEFORE (a time interval specified by the vendor)

WHILE (not all records from S2 have been returned by TD (see notes to tester))

RECEIVE ReadRange-Request,

'Object Identifier' = (L1),

'Property Identifier' = (Log\_Buffer),

'Range' = (any valid range)

TRANSMIT ReadRange-Ack,

'Object Identifier' = (L1),

'Property Identifier' = (Log\_Buffer),

'ItemData' = (a set of records appropriate for this response)

5. MAKE (L1 contain records S1 + S2 + S3)

6. TRANSMIT ConfirmedEventNotification-Request,

'Subscriber Process Identifier' = (any valid value selected for the IUT),

'Initiating Device Identifier' = TD,

'Event Object Identifier' = (the object detecting the event),

'Time Stamp' = (any valid value),

'Notification Class' = (any valid notification class),

'Priority' = (any valid value),

'Event Type' = BUFFER\_READY,

'Notify Type' = ALARM | EVENT,

'AckRequired' = TRUE | FALSE,

'From State' = NORMAL,

'To State' = NORMAL,

'Event Values' = (L1, Log\_Buffer),

(any Unsigned32 value that indicates the end of S2),

(any Unsigned32 value that indicates the addition of S3)

7. RECEIVE Simple-Ack-PDU

8. MAKE (L1 contain records S1 + S2 + S3 + S4 (see note to tester))

9. BEFORE (a time interval specified by the vendor)

WHILE (not all records from S3 have been returned by D1)

RECEIVE ReadRange-Request,

'Object Identifier' = (L1),

'Property Identifier' = (Log\_Buffer),

'Range' = (any valid range)

TRANSMIT ReadRange-Ack,

'Object Identifier' = (L1),

'Property Identifier' = (Log\_Buffer),

'ItemData' = (a set of records appropriate for this response)

10. CHECK (the records from all steps form a contiguous set, including all records in S2, S3, and possibly some portion of S4)

注意事项：在第4步之前，IUT可能已经预读了一些S2。这种情况下，在第4步中未读取这些样本的IUT不会失败。

注意事项：在第8步中添加S4的操作可在第6步之后第10步之前的任何时间进行。

1. 使用任何有效范围读取项目范围响应BUFFER\_READY事件类型的UnconfirmedEventNotification

目的：通过响应BUFFER\_READY事件类型的UnconfirmedEventNotification消息，验证IUT是否可以发起一系列ReadRange服务请求以访问连续的日志记录集。

测试思想、配置要求和测试步骤与9.21.X1中的相同，不同之处在于发出UnconfirmedEventNotification请求而不是ConfirmedEventNotification请求，并且IUT不确认接收到通知。

1. 使用任何有效范围读取项目范围

目的：验证IUT可以使用任何有效范围发起一个或多个ReadRange请求，以访问测试人员指定的日志记录部分。

测试思想：TD包含一个趋势日志对象，该对象具有逻辑上的日志记录集S1。测试人员选择要返回的S1的一部分，并使IUT使用任何有效范围来请求这些记录。然后，测试验证IUT是否可以以与TD返回的记录一致的方式显示记录。

配置要求：TD包含一个日志对象L1，该对象具有一组记录S1。配置IUT为显示返回的日志记录集。

测试步骤：

1. MAKE (L1 contain the set of records S1)

2. MAKE (the IUT request a range of samples from L1)

3. WHILE (not all records from the tester-selected range have been requested)

RECEIVE ReadRange-Request,

'Object Identifier' = (L1),

'Property Identifier' = (Log\_Buffer),

'Range' = (any valid range)

TRANSMIT ReadRange-Ack,

'Object Identifier' = (L1),

'Property Identifier' = (Log\_Buffer),

'ItemData' = (a set of records appropriate for this response)

4. CHECK (the records returned in step 3 include the tester-selected range)

5. MAKE (the IUT display the tester-selected range)

6. CHECK (the records in step 5 are consistent with the records returned in step 3)

注意事项：IUT不需要展示包含日志状态值的记录。

1. 提供包含特定数据类型的日志记录

目的：验证IUT可以发起一个或多个ReadRange请求，使用任何有效范围，访问并显示测试人员指定的具有特定数据类型的日志记录部分。这是用于测试数据表示要求的通用测试。

测试思想：在条款9.21.X3中运行测试，并验证数据表示是否符合被测BIBB指定的标准。

注意事项：由于四舍五入、截断、格式化、语言、转换等原因，IUT呈现的值可能与传输的值不同。

注意事项：IUT不需要展示包含日志状态值的记录。

* 1. 写属性（WriteProperty）服务启动测试

此条款定义了证明IUT可启动WriteProperty服务请求所需的测试。

所需测试:无。

BACnet参考条款:15.9.

1. 写非数组属性

目的:验证IUT是否可以发起不包含“Property Array Index”参数的WriteProperty服务请求。

测试步骤:

1. RECEIVE WriteProperty-Request,

'Object Identifier' = (any valid object identifier),

'Property Identifier' = (any valid non-array property of the specified object),

'Property Value' = (any value appropriate to the specified property)

注意事项:任何由IUT生成的WriteProperty请求都可能有Priority参数。如果包含，参数应在1-16范围内，不包括6。

1. 写数组属性

目的:验证IUT可以发起引用数组属性的特定元素的WriteProperty服务请求。

测试步骤:

1. RECEIVE WriteProperty-Request,

'Object Identifier' = (any valid object identifier),

'Property Identifier' = (any valid array property of the specified object),

'Array Index' = (any valid array index for the specified property),

'Property Value' = (any value appropriate to the specified property)

注意事项:任何由IUT生成的WriteProperty请求都可能有Priority参数。如果有，参数应在1-16范围内，不包括6。

1. 写可命令属性

目的:验证IUT能够发起传递写优先级的WriteProperty服务请求。

测试步骤:

1. RECEIVE WriteProperty-Request,

'Object Identifier' = (any valid object identifier for an object with a commandable property),

'Property Identifier' = (the commandable property of the specified object),

'Property Value' = (any value appropriate to the specified property),

'Priority' = (any unsigned value X: 1 ≤ X ≤ 16)

1. 接受输入和修改属性

目的:这个测试用例验证IUT能够接受用户输入并使用它修改属性。它是一个用于测试数据输入需求的通用测试。

配置:对于这个测试，测试人员应该从对象O1中选择一个属性P1。TD应配置为不支持写多个属性 (WritePropertyMultiple)的执行。

测试步骤:

1. MAKE (the IUT accept a new value for P1 from the user)

2. RECEIVE WriteProperty-Request,

'Object Identifier' = O1,

'Property Identifier' = P1

'Property Value' = (the value provided to the IUT for P1)

3. TRANSMIT BACnet-SimpleACK-PDU

注意事项：由于四舍五入、截断、格式化、语言转换等原因，IUT接受的值可能与传输的值不同。

注意事项:如果被修改的属性是数组元素，IUT可以在步骤2中的WriteProperty-Request中包含一个数组索引Array Index参数。

注意事项:如果IUT还没有确定TD不支持WritePropertyMultiple的执行，IUT可能会启动 WritePropertyMultiple。这种情况下，IUT只有在从TD收到正确的BACnetReject-PDU后自动返回到使用WriteProperty时才能通过测试，这表明不支持WritePropertyMultiple。

1. 接受输入和命令/放弃属性

目的:这个测试用例验证IUT能够接受用户输入并使用它以特定的优先级修改一个可命令的属性。它还测试了IUT在相同优先级时有能力放弃。

配置:对于这个测试，测试人员应该从对象O1中选择一个可命令的属性P1。TD应配置为不支持 WritePropertyMultiple的执行。PR1是将被测试的特定优先级。

测试步骤:

1. MAKE (the IUT accept a new value for P1 from the user, to be commanded at priority PR1)

2. RECEIVE WriteProperty-Request,

'Object Identifier' = O1,

'Property Identifier' = P1,

‘Priority’ = PR1,

'Property Value' = (the value provided to the IUT for P1)

3. TRANSMIT BACnet-SimpleACK-PDU

4. MAKE (the IUT relinquish P1 at priority PR1)

5. RECEIVE WriteProperty-Request,

'Object Identifier' = O1,

'Property Identifier' = P1,

‘Priority’ = PR1,

'Property Value' = NULL

6. TRANSMIT BACnet-SimpleACK-PDU

注意事项：由于四舍五入、截断、格式化、语言转换等原因，IUT接受的值可能与传输的值不同。

注意事项:如果IUT不确定TD不支持WritePropertyMultiple的执行，IUT可能会启动 WritePropertyMultiple。这种情况下，IUT只有在从TD收到正确的BACnetReject-PDU后自动返回到使用WriteProperty时才能通过测试，这表明不支持WritePropertyMultiple。

1. 写数组的长度

目的:验证IUT可以启动WriteProperty服务请求来修改数组的长度。

测试步骤:

1. RECEIVE WriteProperty-Request,

'Object Identifier' = (any valid object identifier),

'Property Identifier' = (any valid array property of the specified object),

‘Array Index’ = 0

'Property Value' = (any unsigned value)

注意事项: IUT生成的任何WriteProperty请求都可能有一个Priority参数。如果有，Priority参数应在1-16范围内，不包括6。

* 1. 写多个属性（WritePropertyMultiple）服务启动测试

该条款定义了证明支持启动WritePropertyMultiple服务请求所需的测试。至少需要支持9.23.2到9.23.4中定义的一种组合，以声明能够启动WritePropertyMultiple服务。

所需测试:无。

BACnet参考条款:15.10。

1. 写单个对象的单个属性

目的:验证IUT可以正确地发起包含单个对象的单个属性的WritePropertyMultiple服务请求。如果IUT不能配置为启动包含单个对象的单个属性的WritePropertyMultiple服务请求，则应省略此测试。

测试步骤:

1. RECEIVE WritePropertyMultiple-Request,

'Object Identifier' = (any valid object identifier),

'Property Identifier' = (any valid non-array property of the specified object),

'Property Value' = (any value appropriate to the specified property)

注意事项: 任何由IUT生成的WritePropertyMultiple请求都可能对所写的任何属性具有Priority参数。如果有，Priority参数应在1-16范围内，不包括6。

1. 写单个对象的多个属性

目的:验证IUT可以正确地发起包含单个对象的多个属性的WritePropertyMultiple服务请求。如果IUT不能配置为启动包含单个对象的多个属性的WritePropertyMultiple服务请求，则应省略此测试。

测试步骤:

1.RECEIVE WritePropertyMultiple-Request,

'Object Identifier' = (any valid object identifier),

'Property Identifier' = (any valid non-array property of the specified object),

'Property Value' = (any value appropriate to the specified property),

'Property Identifier' = (any valid non-array property of the specified object that was not previously used),

'Property Value' = (any value appropriate to the specified property)

-- … (Any number of properties ≥ 2 is acceptable.)

注意事项: 任何由IUT生成的WritePropertyMultiple请求都可能对所写的任何属性具有Priority参数。如果有，Priority参数应在1-16范围内，不包括6。

1. 写多个对象，每个对象一个属性

目的:验证IUT可以正确地启动一个WritePropertyMultiple服务请求，该请求包含多个对象和每个对象的一个属性。如果IUT不能配置为启动一个包含多个对象和每个对象一个属性的 WritePropertyMultiple服务请求，则应省略此测试。

测试步骤:

1. RECEIVE WritePropertyMultiple-Request,

'Object Identifier' = (any valid object identifier),

'Property Identifier' = (any valid non-array property of the specified object),

'Property Value' = (any value appropriate to the specified property),

'Object Identifier' = (any valid object identifier not previously used),

'Property Identifier' = (any valid non-array property of the specified object),

'Property Value' = (any value appropriate to the specified property)

-- … (Any number of (object, property, value) tuples ≥ 2 is acceptable.)

注意事项: 任何由IUT生成的WritePropertyMultiple请求都可能对所写的任何属性具有Priority参数。如果有，Priority参数应在1-16范围内，不包括6。

1. 写多个对象，为每个对象写多个属性

目的:验证IUT可以正确地发起包含多个对象和每个对象的多个属性的WritePropertyMultiple服务请求。如果IUT不能配置为启动包含多个对象和每个对象的多个属性的WritePropertyMultiple服务请求，则应省略此测试。

测试步骤:

1. RECEIVE WritePropertyMultiple-Request,

'Object Identifier' = (any valid object identifier),

'Property Identifier' = (any valid non-array property of the specified object),

'Property Value' = (any value appropriate to the specified property),

'Property Identifier' = (any valid non-array property of the specified object that was not previously used),

'Property Value' = (any value appropriate to the specified property),

-- … (Additional properties may be included here.)

'Object Identifier' = (any valid object identifier not previously used),

'Property Identifier' = (any valid non-array property of the specified object),

'Property Value' = (any value appropriate to the specified property),

'Property Identifier' = (any valid non-array property of the specified object that was not previously used),

'Property Value' = (any value appropriate to the specified property)

-- … (Additional properties may be included here.)

-- … (Additional objects and properties may be included here.)

注意事项: 任何由IUT生成的WritePropertyMultiple请求都可能对所写的任何属性具有Priority参数。如果有，Priority参数应在1-16范围内，不包括6。

1. 写数组属性

目的:验证IUT可以启动WritePropertyMultiple服务请求，该请求引用数组属性的特定元素。

测试步骤:

1. RECEIVE WritePropertyMultiple-Request,

'Object Identifier' = (any valid object identifier),

'Property Identifier' = (any valid array property of the specified object),

'Array Index' = (any valid array index for the specified property),

'Property Value' = (any value appropriate to the specified property),

'Property Identifier' = (any valid non-array property of the specified object),

'Property Value' = (any value appropriate to the specified property)

-- … (Including additional properties, with or without array indexes, is acceptable.)

注意事项: 只需要写一个属性作为数组的一个特定元素，但也可以将多个特定元素作为单个操作来编写。

注意事项: 任何由IUT生成的WritePropertyMultiple请求都可能对所写的任何属性具有Priority参数。如果有，Priority参数应在1-16范围内，不包括6。

1. 写可命令属性

目的:验证IUT可以发起传递写优先级的WritePropertyMultiple服务请求。

测试步骤:

1. RECEIVE WritePropertyMultiple-Request,

'Object Identifier' = (any valid object identifier for an object with a commandable property),

'Property Identifier' = (the commandable property of the specified object),

'Property Value' = (any value appropriate to the specified property),

'Priority' = (any unsigned value X: 1 ≤ X ≤ 16),

'Property Identifier' = (any valid non-array property of the specified object),

'Property Value' = (any value appropriate to the specified property)

注意事项: 只需有一个正在编写的属性是可命令的，但是也可以写入多个可命令属性。

* 1. 设备通信控制（DeviceCommunicationControl）服务启动测试

此条款定义了证明支持启动DeviceCommunicationControl服务请求所需的测试。

所需测试:无。

BACnet参考条款:16.1。

1. 限期(Indefinite Duration), 禁用(Disable), 没有密码(No Password)

目的:验证IUT可以发起DeviceCommunicationControl服务请求，该请求指示通信应该无限期停止，并且不传递密码。

测试步骤:

1. RECEIVE DeviceCommunicationControl-Request,

'Enable/Disable' = DISABLE,

1. 无限期(Indefinite Duration), 禁用(Disable), 有密码(Password)

目的:验证IUT可以发起DeviceCommunicationControl服务请求，该请求指示通信应该无限期停止，并且传递密码。

测试步骤:

1. RECEIVE DeviceCommunicationControl-Request,

'Enable/Disable' = DISABLE,

'Password' = (a password of at least 5 characters)

1. 时间期限(Time Duration), 禁用(Disable), 有密码(Password)

目的:验证IUT可以发起DeviceCommunicationControl服务请求，该请求指示通信应该在特定时间内停止，并传递密码。

测试步骤:

1. RECEIVE DeviceCommunicationControl-Request,

'Time Duration' = (any unsigned value > 0),

'Enable/Disable' = DISABLE,

'Password' = (a password of at least 5 characters)

1. 启用(Enable), 有密码(Password)

目的:验证IUT可以发起DeviceCommunicationControl服务请求，指示应恢复通信并传递密码。

测试步骤:

1. RECEIVE DeviceCommunicationControl-Request,

'Enable/Disable' = ENABLE,

'Password' = (a password of at least 5 characters)

1. 启用(Enable), 没有密码(No Password)

目的:验证IUT可以发起DeviceCommunicationControl服务请求，该请求指示应恢复通信，并且不传递密码。

测试步骤:

1. RECEIVE DeviceCommunicationControl-Request,

'Enable/Disable' = ENABLE,

1. 时间期限(Time Duration), 禁用(Disable), 没有密码(No Password)

目的:验证IUT可以发起DeviceCommunicationControl服务请求，该请求指示通信应该在特定时间内停止，并且不传递密码。如果IUT不支持“no password”选项，则不应执行此测试。

测试步骤:

1. RECEIVE DeviceCommunicationControl-Request,

'Time Duration' = (any unsigned value > 0),

'Enable/Disable' = DISABLE

1. 时间期限(Time Duration), 禁止启动(Disable-Initiation), 有密码(Password)

目的:验证IUT可以发起DeviceCommunicationControl服务请求，该请求指示通信应该在特定时间内停止，并且传递密码。

测试步骤:

1. RECEIVE DeviceCommunicationControl-Request,

'Time Duration' = (any unsigned value in the range from 1 to 65535),

'Enable/Disable' = DISABLE

'Password' = (a password of up to 20 characters)

* 1. 确认专有转换（ConfirmedPrivateTransfer）服务启动测试

所需测试:没有。

BACnet参考条款:16.2。

目的:验证IUT可以启动ConfirmedPrivateTransfer服务。

测试思想:由于专有传输服务按照定义传递非标准服务请求，因此没有测试服务参数和服务过程。该测试只是验证BACnet所需的参数是否得到了正确的传递。

测试步骤:

1. RECEIVE ConfirmedPrivateTransfer-Request,

'Vendor ID' = (any valid vendor identifier),

'Service Number' = (any unsigned value)

* 1. 未确认专有转换（UnconfirmedPrivateTransfer）服务启动测试

所需测试:没有。

BACnet参考条款:16.3。

目的:验证IUT可以启动UnconfirmedPrivateTransfer服务。

测试思想:由于专有传输服务按照定义传递非标准服务请求，因此没有测试服务参数和服务过程。该测试只是验证BACnet所需的参数是否正确传输。

测试步骤:

1. RECEIVE UnconfirmedPrivateTransfer-Reques,

'Vendor ID' = (any valid vendor identifier),

'Service Number' = (any unsigned value)

* 1. 重新初始化设备（ReinitializeDevice）服务启动测试

定义了证明支持启动ReinitializeDevice服务请求所需的测试。

所需测试:无。

BACnet参考条款:16.4。

1. 无密码的冷启动(COLDSTART with no Password)

目的:验证IUT可以发起ReinitializeDevice服务请求，该请求指示应该执行COLDSTART，并且不传递密码。

测试步骤:

1. RECEIVE ReinitializeDevice-Request,

'Reinitialized State of Device' = COLDSTART

2. TRANSMIT BACnet-SimpleACK-PDU

1. 有密码的冷启动(COLDSTART with a Password)

目的:验证IUT可以发起ReinitializeDevice服务请求，该请求指示应该执行COLDSTART，并且传递密码。

测试步骤:

1. RECEIVE ReinitializeDevice-Request,

'Reinitialized State of Device' = COLDSTART,

'Password' = (a password of at least 5 characters)

2. TRANSMIT BACnet-SimpleACK-PDU

1. 无密码的热启动(WARMSTART with no Password)

目的:验证IUT可以发起ReinitializeDevice服务请求，该请求指示应该执行一个WARMSTART，并且不传递密码。

测试步骤:

1. RECEIVE ReinitializeDevice-Request,

'Reinitialized State of Device' = WARMSTART

2. TRANSMIT BACnet-SimpleACK-PDU

1. 有密码的热启动(WARMSTART with a Password)

目的:验证IUT可以发起ReinitializeDevice服务请求，该请求指示应该执行一个WARMSTART，并且传递一个密码。

测试步骤:

1. RECEIVE ReinitializeDevice-Request,

'Reinitialized State of Device' = WARMSTART,

'Password' = (a password of at least 5 characters)

2. TRANSMIT BACnet-SimpleACK-PDU

* 1. 确认文本信息（ConfirmedTextMessage）服务启动测试

此条款定义了证明支持启动ConfirmedTextMessage服务请求所需的测试。

所需测试:无。

BACnet参考条款:16.5。

1. 无报文类的文本报文(Text Message with no Message Class)

目的:验证IUT可以发起不包含“Message Class”参数的ConfirmedTextMessage服务请求。

测试步骤:

1. RECEIVE ConfirmedTextMessage-Request,

'Text Message Source Device' = IUT,

'Message Priority' = NORMAL,

'Message' = (any CharacterString)

2 TRANSMIT BACnet-SimpleACK-PDU 8

1. 无符号报文类的文本报文(Text Message with an Unsigned Message Class)

目的:验证IUT可以启动包含一个无符号的“Message Class”参数的ConfirmedTextMessage服务请求。

测试步骤:

1. RECEIVE ConfirmedTextMessage-Request,

'Text Message Source Device' = IUT,

'Message Class' = (any Unsigned value),

'Message Priority' = NORMAL,

'Message' = (any CharacterString)

2. TRANSMIT BACnet-SimpleACK-PDU

1. 字符串报文类的文本报文(Text Message with a CharacterString Message Class)

目的:验证IUT可以启动ConfirmedTextMessage服务请求，该请求包含一个字符串类型的“Message Class”参数。

测试步骤:

1. RECEIVE ConfirmedTextMessage-Request,

'Text Message Source Device' = IUT,

'Message Class' = (any CharacterString value),

'Message Priority' = NORMAL,

'Message' = (any CharacterString)

2. TRANSMIT BACnet-SimpleACK-PDU

1. 有紧急优先权的文本报文(Text Message with an Urgent Priority)

目的:验证IUT可以发起传递有紧急优先权的ConfirmedTextMessage服务请求。

测试步骤:

1. RECEIVE ConfirmedTextMessage-Request,

'Text Message Source Device' = IUT,

'Message Priority' = URGENT,

'Message' = (any CharacterString)

2. TRANSMIT BACnet-SimpleACK-PDU

* 1. 未确认文本信息（UnconfirmedTextMessage）服务启动测试

此条款定义了证明支持启动UnconfirmedTextMessage服务请求所需的测试

所需测试:无。

BACnet参考条款:16.6。

1. 无报文类的文本报文(Text Message with no Message Class)

目的:验证IUT可以发起不包含“Message Class”参数的UnconfirmedTextMessage服务请求。

测试步骤:

1. RECEIVE UnconfirmedTextMessage-Request,

'Text Message Source Device' = IUT,

'Message Priority' = NORMAL,

'Message' = (any CharacterString)

1. 无符号报文类的文本报文(Text Message with an Unsigned Message Class)

目的:验证IUT可以发起包含无符号的“Message Class”参数的UnconfirmedTextMessage服务请求。

测试步骤:

1. RECEIVE UnconfirmedTextMessage-Request,

'Text Message Source Device' = IUT,

'Message Class' = (any Unsigned value),

'Message Priority' = NORMAL,

'Message' = (any CharacterString)

1. 字符串报文类的文本报文(Text Message with a CharacterString Message Class)

目的:验证IUT可以发起包含字符串“Message Class”参数的UnconfirmedTextMessage服务请求。

测试步骤:

1. RECEIVE UnconfirmedTextMessage-Request,

'Text Message Source Device' = IUT,

'Message Class' = (any CharacterString value),

'Message Priority' = NORMAL,

'Message' = (any CharacterString)

1. 有紧急优先权的文本报文(Text Message with an Urgent Priority)

目的:验证IUT可以发起传递有紧急优先权的UnconfirmedTextMessage服务请求。

测试步骤:

1. RECEIVE TimeSynchronization-Request,

'Text Message Source Device' = IUT,

'Message Priority' = URGENT,

'Message' = (any CharacterString)

* 1. 时间同步（TimeSynchronization）服务启动测试

所需测试:无。

BACnet参考条款:16.7。

目的:验证IUT可以发起TimeSynchronization服务请求。

测试步骤:

1. RECEIVE UTCTimeSynchronization-Request,

'Time' = (the current local date and time)

* 1. UTC时间同步（UTCTimeSynchronization）服务启动测试

所需测试:无。

BACnet参考条款:16.7。

目的:验证IUT可以发起UTCTimeSynchronization service请求。

测试步骤:

1. RECEIVE UTCTimeSynchronization-Request,

'Time' = (the current UTC date and time)

* 1. Who-Has服务启动测试

此条款定义了证明支持初始化Who-Has服务请求所需的测试。至少需要支持9.32.1到9.32.4中定义的服务的一种形式，才能声称能够启动WhoHas服务。

所需测试:无。

BACnet参考条款:16.9。

1. 没有设备实例范围的对象标识符选择

目的:验证IUT可以使用没有设备实例范围的对象标识符形式启动Who-Has服务请求。如果IUT不能发出此表格的Who-Has请求，则省略此测试。

测试步骤:

1. RECEIVE

DESTINATION = LOCAL BROADCAST | GLOBAL BROADCAST,

SOURCE = IUT,

Who-Has-Request,

'Object Identifier' = (any object identifier)

1. 没有设备实例范围的对象名称选择

目的:验证IUT可以使用没有设备实例范围的对象名称形式启动Who-Has服务请求。如果IUT不能发出此表格的Who-Has请求，则省略此测试。

测试步骤:

1. RECEIVE

DESTINATION = LOCAL BROADCAST | GLOBAL BROADCAST,

SOURCE = IUT,

Who-Has-Request,

'Object Name' = (any CharacterString)

1. 使用设备实例范围的对象标识符选择

目的:验证IUT可以使用带有设备实例范围的对象标识符形式启动Who-Has服务请求。如果IUT不能发出此表格的Who-Has请求，则省略此测试。

测试步骤:

1. RECEIVE

DESTINATION = LOCAL BROADCAST | GLOBAL BROADCAST,

SOURCE = IUT,

Who-Has-Request,

'Device Instance Range Low Limit' = (any integer X: 1 ≤ X ≤ 'Device Instance Range High Limit'),

'Device Instance Range High Limit' = (any integer Y: 'Device Instance Range Low Limit' ≤ Y ≤ 4,194,303),

'Object Identifier' = (any object identifier)

1. 使用设备实例范围的对象名称选择

目的:验证IUT可以使用带有设备实例范围的对象名称形式初始化Who-Has服务请求。如果IUT不能发出此形式的Who-Has请求，则省略此测试。

测试步骤:

1. RECEIVE

DESTINATION = LOCAL BROADCAST | GLOBAL BROADCAST,

SOURCE = IUT,

Who-Has-Request,

'Device Instance Range Low Limit' = (any integer X: 1 ≤ X ≤ 'Device Instance Range High Limit'),

'Device Instance Range High Limit' = (any integer Y: 'Device Instance Range Low Limit' ≤ Y ≤ 4,194,303),

'Object Name' = (any CharacterString)

* 1. I-Have服务启动测试

在9.32中，Who-Has服务执行测试涵盖了启动I-Have服务请求能力的验证。

* 1. Who-Is服务启动测试

此条款定义了用于证明支持启动Who-Is服务请求所必需的测试。至少需要支持一种9.34.1与9.34.2定义的服务形式，以声称能够启动Who-Is服务。

所需测试:无。

BACnet参考条款:16.10。

1. 无设备实例范围的Who-Is请求

目的:验证IUT是否可以启动在无设备实例范围的Who-Is服务请求。如果IUT不能发出此形式的Who-Is请求，则省略此测试。

测试步骤:

1. RECEIVE

DESTINATION = LOCAL BROADCAST | GLOBAL BROADCAST,

SOURCE = IUT,

Who-Is-Request

1. 有设备实例范围的Who-Is请求

目的: 验证IUT是否可以启动在有设备实例范围的Who-Is服务请求。如果IUT不能发出此形式的Who-Is请求，则省略此测试。

测试步骤:

1. RECEIVE

DESTINATION = LOCAL BROADCAST | GLOBAL BROADCAST,

SOURCE = IUT,

Who-Is-Request,

'Device Instance Range Low Limit' = (any integer X: 1 ≤ X ≤ 'Device Instance Range High Limit'),

'Device Instance Range High Limit' = (any integer Y: 'Device Instance Range Low Limit' ≤ Y ≤ 4,194,303)

1. 无设备实例范围的Who-Is请求

目的: 验证IUT是否可以启动在无设备实例范围的单播Who-Is服务请求。如果IUT不能发出此格式的Who-Is请求，则省略此测试。

测试步骤:

1. RECEIVE

DESTINATION = TD,

SOURCE = IUT,

Who-Is-Request

* 1. I-Am服务启动测试

10.30中的Who-Is服务执行测试涵盖了启动I-Am服务请求能力的验证。

* 1. VT-Open服务启动测试

此条款定义了证明支持启动VT-Open服务请求所需的测试。

所需测试: ReadProperty服务执行测试，10.18。

BACnet参考条款:17.2和12.11。

1. 默认终端VT-class

目的：验证IUT是否可以为DEFAULT\_TERMINAL VT-class启动VT-open服务请求，以及打开的VT会话是否反映在IUT设备对象的Active\_VT\_Sessions属性中。

配置要求：IUT的配置应确保没有活动状态的VT会话。

测试步骤：

1.RECEIVE VT-Open-Request,

'VT-class' = DEFAULT\_TERMINAL,

'Local VT Session Identifier' = (any valid session identifier, SIUT)

2.TRANSMIT VT-Open-ACK,

'Remote VT Session Identifier' = (any valid session identifier, STD)

3. VERIFY (the IUT's Device object), Active\_VT\_Sessions = (the newly established session)

如果IUT能够有多个活动状态的VT会话，则测试人员可以选择将9.36.2中进行的测试所创建的会话保持为活动状态。否则，应该关闭VT会话。这可以作为验证启动VT-close服务请求（9.37）或执行VT-Close服务（10.35.1）的测试的一部分完成。

1. 其他VT-class

目的：验证IUT是否可以为所有支持的可选VT-class启动VT-open服务请求，以及开放的VT会话是否反映IUT设备对象的Active\_VT\_Sessions属性。如果IUT不支持除DEFAULT\_TERMINAL以外的VT-class，则省略该测试。

测试思想：为每个受支持的VT-class打开一个新的VT会话。在所有受支持的VT-class完成测试、或在达到支持的最大打开会话数之前，先前打开的会话保持打开状态。如果所有VT-class完成测试之前达到了打开会话的最大数目，那么会话在继续下一个VT-class之前关闭。IUT或TD均可关闭会话。

测试步骤：

1. REPEAT X = (all supported VT classes except DEFAULT\_TERMINAL) DO {

RECEIVE VT-Open-Request,

'VT-class' = X,

'Local VT Session Identifier' = (any valid unique session identifier, SIUT,i)

TRANSMIT VT-Open-ACK,

'Remote VT Session Identifier' = (any valid unique session identifier, STD,i)

VERIFY (the IUT's Device object),

Active\_VT\_Sessions = (a list including the newly established session and any sessions previously established as part of this test but not yet closed)

IF (the IUT cannot support additional active VT sessions) THEN

(TRANSMIT VT-Close-Request,

'List of Remote VT Session Identifiers' = (the IUT's session identifier for the most recently opened session)

RECEIVE BACnet-SimpleACK-PDU

)

|

(RECEIVE VT-Close-Request,

'List of Remote VT Session Identifiers' = (the TD's session identifier for the most recently opened session)

TRANSMIT BACnet-SimpleACK-PDU

)

}

* 1. VT-Close服务启动测试

本条款定义了证明支持启动VT-Close服务请求所必需的测试。IUT应通过9.37.1中规定的试验。如果IUT支持多个活动状态的VT会话，它还应通过9.37.2和9.37.3中规定的测试。

所需测试：ReadProperty服务执行测试，10.18。

BACnet参考条款：17.3和12.11。

1. 关闭单个打开的VT对话

目的：验证IUT是否可以启动一个VT-Close服务请求，该请求传递单个打开的VT会话的会话标识符，并且一旦关闭，该会话将从IUT设备对象的Active\_VT\_Sessions属性中删除。

测试思想：在测试开始时，IUT有一个活动状态的VT会话。IUT试图通过启动VT-Close服务请求来关闭该活动会话。TD确认VT-Close请求，然后读取Active\_VT\_Sessions属性以验证它是否已正确更新。

配置要求：IUT应配置一个活动状态的VT会话。如果IUT刚完成测试8.36.1，并且会话随后没有关闭，那么IUT将处于这种状态。

测试步骤：

1. VERIFY (the IUT's Device object), Active\_VT\_Sessions = (a single valid session)

2. RECEIVE VT-Close-Request,

'List of Remote VT Session Identifiers' = (a single session identifier, STD, appropriate for the TD's view of the open session)

3. TRANSMIT BACnet-SimpleACK-PDU

4. VERIFY (the IUT's Device object), Active\_VT\_Sessions = (an empty list)

1. 关闭多个打开的VT对话中的一个

目的：验证IUT是否可以启动一个VT-Close服务请求，该请求从多个打开的VT会话传递一个会话标识符，并且一旦关闭，该会话将从IUT设备对象的Active\_VT\_Sessions属性中删除。

测试思想：在测试开始时，IUT有多个活动状态的VT会话。IUT试图通过启动VT-Close服务请求来关闭一个活动状态的会话。TD确认VT-Close请求，然后读取Active\_VT\_Sessions属性，以验证是否预期会话已关闭，但其他会话仍保留。

配置要求：IUT应配置一个激活的VT会话。如果IUT刚完成测试9.36.2，并且会话随后没有关闭，那么IUT将处于这种状态。

测试步骤：

1. VERIFY (the IUT's Device object), Active\_VT\_Sessions = (a list of two or more valid sessions)

2. RECEIVE VT-Close-Request,

'List of Remote VT Session Identifiers' = (a single session identifier, STD, appropriate for the TD's view of one of the open sessions)

3. TRANSMIT BACnet-SimpleACK-PDU

4. VERIFY (the IUT's Device object), Active\_VT\_Sessions = (the same list as in step 1 except that the closed session is no longer present)

1. 关闭多个打开的VT对话

目的：验证IUT是否可以启动一个VT-Close服务请求，该请求从多个打开的VT会话传递多个会话标识符，并且一旦关闭，会话将从IUT设备对象的Active\_VT\_Sessions属性中删除。

测试思想：在测试开始时，IUT有多个活动状态的VT会话。IUT试图通过启动单个VT-Close服务请求来关闭这些活动会话中的多个会话。TD确认VT-Close请求，然后读取Active\_VT\_Sessions属性，以验证是否预期会话已关闭，但其他会话仍保留。

配置要求：IUT应配置多个活动VT会话。如果IUT刚刚完成测试8.36.2，并且开放会话随后未关闭，则IUT将处于这种状态。

测试步骤：

1. VERIFY (the IUT's Device object), Active\_VT\_Sessions = (a list of two or more valid sessions)

2. RECEIVE VT-Close-Request,

'List of Remote VT Session Identifiers' = (two or more session identifiers, STD, i and STD,j, appropriate for the TD's view of the open sessions)

3. TRANSMIT BACnet-SimpleACK-PDU

4. VERIFY (the IUT's Device object), Active\_VT\_Sessions = (the same list as in step 1 except that the closed sessions are no longer present)

* 1. VT-Data服务初始化测试

本条款定义了证明支持启动VT-Data服务请求所必需的测试。提供方法令IUT传输足够长的VT数据流，以验证“VT Data Flag”的正确顺序。这一条件的满足属于本地问题。如果IUT为虚拟终端会话的虚拟操作员接口端，则应用测试9.38.1。否则，应用测试9.38.2。

所需测试：VT-Open服务启动测试，9.36；VT-Open服务执行测试，10.34；VT-Close服务执行测试, 10.35.

BACnet参考条款：17.4。

1. 虚拟操作者界面(Virtual Operator Interface)

目的：用于验证IUT是否将“VT-data Flag”初始化为零，并在此会话的每个新VT-Data请求中将值在0和1之间交替。它还验证了如果只接受一部分数据时，IUT会正确对数据进行排序。

测试思想：使用TD建立虚拟终端会话。会话需要足够长的时间来验证VT- data标志的顺序。在会话的某一点上，TD将只接受部分数据，以验证IUT是否正确地对数据重新排序，以便下次传输。所有这些都完成后，TD将终止会话

测试步骤：

1. RECEIVE VT-Open-Request,

'VT-class' = DEFAULT\_TERMINAL,

'Local VT Session Identifier' = (any valid session identifier, SIUT)

2. TRANSMIT VT-Open-ACK,

'Remote VT Session Identifier' = (any valid session identifier, STD)

3. MAKE (IUT启动VT-Data-Request)

4. RECEIVE VT-Data-Request,

'VT-session Identifier' = STD,

VT-new Data' = (any valid DEFAULT\_TERMINAL data more than one character in length),

'VT-data Flag' = 0

5. TRANSMIT VT-Data-ACK,

'All new Data Accepted' = FALSE,

'Accepted Octet Count' = 1

6. RECEIVE VT-Data-Request,

'VT-session Identifier' = STD,

'VT-new Data' = (a continuation of the DEFAULT\_TERMINAL data beginning with the second character in step 4),

'VT-data Flag' = 1

7. TRANSMIT VT-Data-ACK,

'All new Data Accepted' = TRUE

8. MAKE (the IUT initiate a VT-Data-Request)

9. RECEIVE VT-Data-Request,

'VT-session Identifier' = STD,

'VT-new Data' = (any valid DEFAULT\_TERMINAL data),

'VT-data Flag' = 0

10. TRANSMIT VT-Data-ACK,

'All new Data Accepted' = TRUE

11. TRANSMIT VT-Close-Request,

'List of VT Session Identifiers' = SIUT

12. RECEIVE BACnet-SimpleACK-PDU

1. 虚拟终端(Virtual Terminal)

目的：用于验证IUT是否将“VT-data Flag”初始化为零，并在此会话的每个新VT-Data请求中将值在0和1之间交替。它还验证了如果只接受一部分数据时，IUT会正确对数据进行排序。

测试思想：使用TD建立虚拟终端会话。会话需要足够长的时间来验证VT-data标志的顺序。在会话的某一点上，TD将只接受部分数据，以验证IUT是否正确地对数据重新排序，以便下次传输。所有这些都完成后，TD将终止会话。

测试步骤：

1. TRANSMIT VT-Open-Request,

'VT-class' = DEFAULT\_TERMINAL,

'Local VT Session Identifier' = (any valid session identifier, STD)

2. RECEIVE VT-Open-ACK,

'Remote VT Session Identifier' = (any valid session identifier, SIUT)

3. TRANSMIT VT-Data-Request,

'VT-session Identifier' = SIUT,

'VT-new Data' = (any data that will trigger the IUT to transfer a VT data stream),

'VT-data Flag' = 0

4. RECEIVE VT-Data-ACK,

'All new Data Accepted' = TRUE

5. RECEIVE VT-Data-Request,

'VT-session Identifier' = STD,

'VT-new Data' = (any valid DEFAULT\_TERMINAL data more than one character in length),

'VT-data Flag' = 0

6. TRANSMIT VT-Data-ACK,

'All new Data Accepted' = FALSE,

'Accepted Octet Count' = 1

7. RECEIVE VT-Data-Request,

'VT-session Identifier' = STD,

'VT-new Data' = (a continuation of the DEFAULT\_TERMINAL data beginning with the second character in step 5),

'VT-data Flag' = 1

8. TRANSMIT VT-Data-ACK,

'All new Data Accepted' = TRUE

9. MAKE (the IUT initiate a VT-Data-Request)

10. RECEIVE VT-Data-Request,

'VT-session Identifier' = STD,

'VT-new Data' = (any valid DEFAULT\_TERMINAL data),

'VT-data Flag' = 0

11. TRANSMIT VT-Data-ACK,

'All new Data Accepted' = TRUE

12. TRANSMIT VT-Close-Request,

'List of VT Session Identifiers' = SIUT

13. RECEIVE BACnet-SimpleACK-PDU

* 1. 请求密钥（RequestKey）服务初始化测试

此条款定义了证明启动RequestKey服务请求能力所需的测试。

必须提供装置，使IUT向TD-密钥服务器发送RequestKey服务请求，并使用由此获得的会话密钥（SK）与另一个设备通信。

此测试中，所有指定要加密的PDU首先被填充，然后使用指定的密钥加密。

配置要求：IUT应配置TD已知的专有会话密钥，PK。还应将IUT配置为已知TD可接受的最大APDU长度，以便其加密的APDU可根据需要进行填充。

1. 初步测试

所需测试：无。

BACnet参考条款：24.4.

目的：验证IUT是否可以向密钥服务器发出RequestKey服务请求，并用其PK进行填充和加密。

测试步骤：

1.RECEIVE RequestKey-Request,

'Requesting Device Identifier' = IUT,

'Requesting Device Address' = (the BACnetAddress of the IUT),

'Remote Device Identifier' = (any device identifier known to the TD key server),

'Remote Device Address' = (the BACnetAddress corresponding to the remote device)

注：本PDU的服务请求部分应使用PKIUT加密。

1. 随机填充测试

所需测试：初步测试，9.39.1。

BACnet参考条款：24.1.4。

目的：验证IUT是否发出关于随机填充的RequestKey服务请求。

测试步骤：此测试用例的测试步骤与9.39.1中的测试步骤相同。

注意事项：本测试用例的通过结果与9.39.1中的测试步骤相同，只是用于填充IUT传输的APDU的字节序列与9.39.1中观察结果不同。

* 1. 验证服务启动测试

本条款定义了证明在五个条件下启动Authenticate服务请求的能力所必需的测试。它们是对等身份验证（Peer Authentication）,消息执行身份验证（Message Execution Authentication）, 消息启动身份验证（Message Initiation Authentication）, 操作者身份验证（Operator Authentication）和加密会话（Enciphered Session）。并非所有支持Authentication服务的设备都支持所有启动模式；只需要执行适用的测试。

此测试中指定要加密的所有PDU首先被填充，然后用指定的密钥进行加密。

配置要求：IUT应配置一个TD已知的专用56位加密密钥。

1. 对等身份验证(Peer Authentication)

所需测试：无。

BACnet参考条款：24.2.2和24.5。

目的：验证设备具有正确启动Authenticate服务请求并实现对等身份验证的能力。

配置要求：IUT应配置一个专用的56位密码密钥。TD和IUT之间进行安全会话之前应建立会话密钥SKTD,IUT。这可以通过执行10.38.1中的测试来完成。

测试步骤：

1. RECEIVE Authenticate-Request,

'Pseudo Random Number' = (any valid pseudo random number)

2. TRANSMIT Authenticate-Request-ACK,

'Modified Random Number' = (the modified pseudo random number)

1. 消息执行身份验证(Message Execution Authentication)

所需测试：ReadProperty服务执行测试, 10.18.

BACnet参考条款：24.2.3和24.5。

目的：验证设备正确启动用于实现消息执行身份验证的Authenticate服务请求能力。

测试思想：在TD和IUT之间建立了一个安全会话。IUT使用经过Authenticate服务执行过程发出ReadProperty请求。

配置要求：IUT应配置一个专用的56位密码密钥。TD和IUT之间进行安全会话之前应建立会话密钥SKTD,IUT。这可以通过执行10.38.1中的测试来完成。

测试步骤：

1. RECEIVE Authenticate-Request,

'Pseudo Random Number' = (any valid pseudo random number),

'Expected Invoke ID' = (any valid invoke ID)

注：本PDU的服务请求部分应使用SKTD,IUT进行加密。

2. RECEIVE ReadProperty-Request,

Invoke ID = (the 'Expected Invoke ID' used in step 1),

'Object Identifier' = (any valid object identifier),

'Property Identifier' = (any supported property of the specified object)

3. TRANSMIT Authenticate-Request-ACK,

'Modified Random Number' = (the modified pseudo random number)

注：本PDU的服务请求部分应使用SKTD,IUT进行加密。

4. TRANSMIT ReadProperty-ACK,

'Object Identifier' = (the object identifier used in step 2),

'Property Identifier' = (the property identifier used in step 2),

'Property Value' = (any valid value for the specified property)

1. 消息启动身份验证(Message Initiation Authentication)

此条款定义了必要的测试，用于证明支持执行Authenticate服务以进行消息启动身份验证。

1. 通过密钥服务器Key-Server的消息初始化身份验证

目的：验证是否能够正确启动Authenticate服务请求以实现消息启动身份验证。如果IUT不是密钥服务器，则省略该测试。

所需测试：无。

BACnet参考条款：24.2.1（a，b），24.2.4。

配置要求：IUT应配置与TD对应的私有56位密钥。

测试步骤：

1. TRANSMIT RequestKey-Request,

'Requesting Device Identifier' = IUT,

'Requesting Device Address' = (IUT的BACnetAddress),

'Remote Device Identifier' = (any device identifier known to the TD key server),

'Remote Device Address' = (the BACnetAddress corresponding to the remote device)

注：本PDU的服务请求部分应使用PKTD进行加密。

2. RECEIVE Authenticate-Request,

'Pseudo Random Number' = (any valid pseudo random number),

注：本PDU的服务请求部分应使用PKTD进行加密。

1. 消息启动身份验证，点对点

所需测试：ReadProperty服务执行测试，10.18。

BACnet参考条款：24.2.4和24.5。

目的：验证是否能够正确启动Authenticate服务请求以实现消息启动身份验证。如果IUT只是一个密钥服务器，则省略该测试。

测试思想：在TD和IUT之间建立了一个安全会话。TD向ITT发出应用服务请求，然后IUT验证该请求的来源。

配置要求：IUT应配置一个私有56位密码密钥。TD和IUT之间使用会话密钥进行安全会话之前应建立SKTD,IUT。这可以通过执行10.38.1中的测试来完成。

测试步骤：

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = (any object supported by the IUT),

'Property Identifier' = (any property of the specified object)

2. RECEIVE Authenticate-Request

'Pseudo Random Number' = (any valid pseudo random number),

'Expected Invoke ID' = (the invoke ID used in step 1)

注：本PDU的服务请求部分应使用SKTD,IUT。

3. TRANSMIT Authenticate-Request-ACK,

'Modified Random Number' = (the modified pseudo random number)

注：本PDU的服务请求部分应使用SKTD,IUT。

4. RECEIVE ReadProperty-ACK,

'Object Identifier' = (the object specified in step 1),

'Property Identifier' = (the property specified in step 1),

'Property Value' = (the value of the specified property as indicated by the EPICS)

1. 操作者身份验证(Operator Authentication)

所需测试：无。

BACnet参考条款：24.2.5。

目的：验证是否能够正确启动Authenticate服务请求以实现操作着身份验证。如果IUT仅为密钥服务器或不支持操作员身份验证，则应省略此测试。

测试思想：TD执行密钥服务器功能，包含操作员密码列表。IUT验证操作员和密码。

配置要求：IUT应配置一个专用的56位密码密钥。

测试步骤：

1. RECEIVE Authenticate-Request

'Pseudo Random Number' = (any valid pseudo random number),

'Operator Name' = (any CharacterString indicating the name of the operator),

'Operator Password' = (any CharacterString indicating the password)

注：该PDU的服务请求部分应使用加密PKIUT。

1. 加密会话(Enciphered Session)

所需测试：ReadProperty服务启动测试，8.18。

BACnet参考条款：24.3.1、24.3.2和24.5。

目的：验证IUT正确启动Authenticate服务请求以启动和终止加密会话的能力。

测试思想：IUT通过发送一个Authenticate请求来启动与TD的加密会话。TD验证请求并启动会话。然后，IUT从TD的设备对象读取属性。然后，IUT尝试通过发送另一个Authenticate请求来终止会话。需要注意，此测试中所有消息的服务请求部分都是使用SKTD,IUT进行加密的。

配置要求：IUT应配置一个私有的56位密码密钥。TD和OUT之间使用会话密钥进行安全会话之前应建立SKTD,IUT。这可以通过执行10.38.1中的测试来完成。

测试步骤：

1. RECEIVE Authenticate-Request,

'Pseudo Random Number' = (any valid pseudo random number),

'Start Enciphered Session' = TRUE

2. TRANSMIT Authenticate-Request,

'Pseudo Random Number' = (any valid pseudo random number),

'Expected Invoke ID' = (the invoke ID used in step 1)

3. RECEIVE Authenticate-ACK,

'Modified Random Number' = (the modified random number from step 2)

4. TRANSMIT Authenticate-ACK,

'Modified Random Number' = (the modified random number from step1)

注：此时将启动加密会话。

5. MAKE (the IUT read a property from the Device object of the TD)

6. RECEIVE ReadProperty-Request,

'Object Identifier' = (the Device object of the TD),

'Property Identifier' = (any supported property)

7. TRANSMIT ReadProperty-ACK,

'Object Identifier' = (the object specified in step 6),

'Property Identifier' = (the property specified in step 6),

'Property Value' = (the value of the specified property)

8. RECEIVE Authenticate-Request,

'Pseudo Random Number' = (any valid pseudo random number),

'Start Enciphered Session' = FALSE

9. TRANSMIT Authenticate-Request,

'Pseudo Random Number' = (any valid pseudo random number),

'Expected Invoke ID' = (the invoke ID used in step 8)

10. RECEIVE Authenticate-ACK,

'Modified Random Number' = (the modified random number from step 9)

11. TRANSMIT Authenticate-ACK,

'Modified Random Number' = (the modified random number from step 8)

1. 应用程序服务执行测试的一般规定

本章节中定义的测试用例应用于验证BACnet设备是否正确地实现了指定应用程序服务的服务过程。应测试BACnet设备是否正确执行PICS中指示支持的每个应用程序服务。

对于本章节中包含的每个应用程序服务都定义了几个测试用例共同测试在BACnet标准中为服务定义的各种选项和特性。测试用例是在IUT和TD之间交换的一个或多个消息序列，以确定特定选项或特性是否正确实现。将具有类似或相关目的的多个测试用例收集到测试组中。

在某些情况下，IUT可能无法证明与特定测试用例的一致性，因为该测试适用于需要特定BACnet对象或IUT不支持的可选属性的功能。例如，设备可能支持文件访问服务，但仅限于流访问文件。这样的设备将无法证明它可以实现文件访问服务的记录访问功能。当这种情况发生时，如果PICS文件明确指出了限制，则应认为IUT符合BACnet。未记录该限制则不符合BACnet标准。除非由于不受支持的对象或不受支持的可选属性而发生冲突，否则应支持BACnet应用程序服务的所有功能和可选参数。

对于每个应用程序服务，测试分为两种类型，正响应测试和负响应测试。正响应测试验证了IUT能够正确处理预期成功完成服务的情况。负响应测试验证了对可能发生的各种错误情况的正确处理。负响应测试包括不适当的服务参数，但不包括编码错误或PDU格式不正确的情况。14.4中定义了确保IUT能够处理异常PDU的测试。

许多测试用例允许在服务参数中使用值的灵活性。测试人员可以自由选择测试用例中定义的约束内的任何值。IUT应能正确响应测试人员可能做出的任何有效选择。EPICS作为明确的参考，表明支持的BACnet功能和对象数据库的配置。BACnet功能或EPICS中定义的对象数据库中的属性值与为测试用例定义的消息中返回的值之间的任何差异都导致测试失败。例如，如果测试步骤涉及读取数据库中对象的属性，则返回的值必须与EPICS中提供的值匹配。

* 1. 确认警报（AcknowledgeAlarm）服务执行测试

本节定义了验证支持执行确认报警服务请求所需的测试项。

所需测试：ReadProperty服务执行测试，10.18。

BACnet参考条款：13.5。

支持启动ConfirmedEventNotification服务请求的BACnet设备应通过10.1.1.1-10.1.1.3和10.1.2.1-10.1.2.4中的测试项。支持启动UnconfirmedEventNotification服务请求的BACnet设备应通过10.1.1.4-10.1.1.6和10.1.2.5-10.1.2.7中的测试项。

1. AcknowledgeAlarm正响应执行测试

此测试组的目的是在预期服务成功完成的情况下验证AcknowledgeAlarm服务请求的正确执行。

1. 使用Time格式的“Time of Acknowledgment”参数时，确认事件通知的成功确认警报

目的：验证ConfirmedEventNotification所发出警报的成功确认，包括通知其他工作站和更新Acked\_Transitions状态。使用Time格式的“Time of Acknowledgment”参数。

测试思想：触发警报使IUT通知TD和至少一个其他设备。TD确认警报并验证IUT是否正确记录了确认。IUT通知所有其他接受者警报已被确认。

配置要求：IUT应配置至少一个可以检测报警条件并发送确认通知的对象。Acked\_Transitions属性应具有值B'111'，表示已确认所有转换。TD和至少一个其他BACnet设备应是警报通知的接收者。

测试步骤：

1. MAKE(a change that triggers the detection of an alarm event in the IUT)

2. WAIT (Time\_Delay)

3. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object detecting the alarm),

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (any valid event type),

'Notify Type' = (the notify type configured for this event),

'AckRequired' = TRUE,

'From State' = NORMAL,

'To State' = (any appropriate non-normal event state),

'Event Values' = (the values appropriate to the event type)

4. TRANSMIT BACnet-SimpleACK-PDU

5. RECEIVE

DESTINATION = (at least one device other than the TD),

SOURCE = IUT,

ConfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object detecting the alarm),

'Time Stamp' = (the timestamp or sequence number received in step 3),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (any valid event type),

'Notify Type' = (the notify type configured for this event),

'AckRequired' = TRUE,

'From State' = NORMAL,

'To State' = (any appropriate non-normal event state),

'Event Values' = (the values appropriate to the event type)

6. TRANSMIT BACnet-SimpleACK-PDU

7. VERIFY (the 'Event Object Identifier' from the event notification), Acked\_Transitions = (FALSE, TRUE, TRUE)

8. TRANSMIT AcknowledgeAlarm-Request,

'Acknowledging Process Identifier' = (the value of the 'Process Identifier' parameter in the event notification),

'Event Object Identifier' = (the 'Event Object Identifier' from the event notification),

'Event State Acknowledged' = (the state specified in the 'To State' parameter of the notification),

'Time Stamp' = (the time stamp conveyed in the notification),

'Time of Acknowledgment' = (the TD’s current time using a Time format)

9. RECEIVE BACnet-Simple-ACK-PDU

10. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object detecting the alarm),

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (the event type included in step 3),

'Notify Type' = ACK\_NOTIFICATION,

'To State' = (the 'To State' used in step 3)

ELSE

BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object detecting the alarm),

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (the event type included in step 3),

'Notify Type' = ACK\_NOTIFICATION,

'To State' = (the 'To State' used in step 3)

11. TRANSMIT BACnet-SimpleACK-PDU

12. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

BEFORE **Notification Fail Time**

RECEIVE

DESTINATION = (at least one device other than the TD),

SOURCE = IUT,

ConfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object detecting the alarm),

'Time Stamp' = (the timestamp or sequence number received in step 10),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (the event type included in step 3),

'Notify Type' = ACK\_NOTIFICATION,

'To State' = (the 'To State' used in step 3)

ELSE

BEFORE **Notification Fail Time**

RECEIVE

DESTINATION = (at least one device other than the TD),

SOURCE = IUT,

ConfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object detecting the alarm),

'Time Stamp' = (the timestamp or sequence number received in step 10),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (the event type included in step 3),

'Notify Type' = ACK\_NOTIFICATION

13. TRANSMIT BACnet-SimpleACK-PDU

14. VERIFY (the 'Event Object Identifier' from the event notification), Acked\_Transitions = (TRUE,TRUE,TRUE)

注意事项：步骤12中用于确认通知的目的地址应与步骤5中使用的地址相同。在协议版本1协议修订版1中的确认通知中添加了“To State”参数。在此之前版本的实现不包含此参数。当一个特定事件需要多个事件通知时，IUT发送它们的顺序是不相关的。

1. 使用序号格式的“Time of Acknowledgment”参数时，确认事件通知的成功确认警报

目的：验证ConfirmedEventNotification所发出的警报的成功确认，包括通知其他工作站和更新Acked\_Transitions状态。使用序号格式的“Time of Acknowledgment”参数。

测试思想：触发警报使IUT通知TD和至少一个其他设备。TD确认警报并验证IUT是否正确记录了确认。IUT通知所有其他接受者警报已被确认。

配置要求：IUT应配置至少一个可以检测报警条件并发送确认通知的对象。Acked\_Transitions属性应具有值B'111'，表示已确认所有转换。TD和至少一个其他BACnet设备应是警报通知的接收者。

测试步骤：应遵循10.1.1.1中定义的测试步骤，但确认报警服务请求的“Time of Acknowledgment”参数应为序号。

注意事项：测试通过的结果与10.1.1.1中描述的测试通过消息序列相同。

1. 使用Date Time格式的Time of Acknowledgment参数，确认事件通知的成功确认警报

目的：验证ConfirmedEventNotification所发出警报的成功确认，包括通知其他工作站和更新Acked\_Transitions状态。使用Date Time格式的“Time of Acknowledgment”参数。

测试思想：触发警报使IUT通知TD和至少一个其他设备。TD确认警报，并验证IUT是否正确记录了确认。IUT通知所有其他收件人警报已被确认。

配置要求：IUT至少配置一个对象，能够检测到报警条件并发送确认通知。Acked\_Transitions属性的值应为B'111'，表示已确认所有转换。TD和至少一个其他BACnet设备应为警报通知的接收者。

测试步骤：应遵循10.1.1.1中定义的测试步骤，除了AcknowledgeAlarm服务请求的“Time of Acknowledgment”参数应使用BacnetDateTime格式来传输当前的时间。

注意事项：测试通过的结果与10.1.1.1中描述的测试通过消息序列相同。

1. 使用Time格式的“Time of Acknowledgment”参数，非确认事件通知的成功确认警报

目的：验证是否成功确认由UnconfirmedEventNotification发出的警报，包括通知其他工作站和更新Acked\_Transitions状态。使用Time格式的“Time of Acknowledgment”参数。

测试思想：触发警报使IUT通知TD和至少一个其他设备。TD确认警报，并验证IUT是否正确记录了确认。IUT通知所有其他收件人警报已被确认。

配置要求：IUT至少配置一个对象，可以检测报警条件并发送非确认通知。Acked\_Transitions属性的值应为B'111'，表示所有的转换都已被确认。TD和至少一个其他BACnet设备应该是警报通知的接收者。

测试步骤：

1. MAKE (a change that triggers the detection of an alarm event in the IUT)

2. WAIT (Time\_Delay)

3. BEFORE **Notification Fail Time**

RECEIVE UnconfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object detecting the alarm),

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event type),

'Event Type' = (any valid event type),

'Notify Type' = (the notify type configured for this event),

'AckRequired' = TRUE,

'From State' = NORMAL,

'To State' = (any appropriate non-normal event state),

'Event Values' = (the values appropriate to the event type)

4. IF (the notification in step 3 was not a broadcast)THEN

RECEIVE

DESTINATION = (at least one device other than the TD),

SOURCE = IUT,

UnconfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object detecting the alarm),

'Time Stamp' = (the timestamp or sequence number received in step 3),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event type),

'Event Type' = (any valid event type),

'Notify Type' = (the notify type configured for this event),

'AckRequired' = TRUE,

'From State' = NORMAL,

'To State' = (any appropriate non-normal event state),

'Event Values' = (the values appropriate to the event type)

5. VERIFY (the 'Event Object Identifier' from the event notification), Acked\_Transitions = (FALSE,TRUE,TRUE)

6. TRANSMIT AcknowledgeAlarm-Request,

'Acknowledging Process Identifier' = (the value of the 'Process Identifier' parameter in the event notification),

'Event Object Identifier' = (the 'Event Object Identifier' from the event notification),

'Event State Acknowledged' = (the state specified in the 'To State' parameter of the notification),

'Time Stamp' = (the time stamp conveyed in the notification),

'Time of Acknowledgment' = (the TD’s current time using a Time format)

7. RECEIVE BACnet-Simple-ACK-PDU

8. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

BEFORE **Notification Fail Time**

RECEIVE

DESTINATION = LOCAL BROADCAST | GLOBAL BROADCAST | TD,

SOURCE = IUT,

UnconfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object detecting the alarm),

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event type),

'Event Type' = (any valid event type),

'Notify Type' = ACK\_NOTIFICATION,

'To State' = (the 'To State' used in step 3 or 4)

ELSE

BEFORE **Notification Fail Time**

RECEIVE

DESTINATION = LOCAL BROADCAST | GLOBAL BROADCAST | TD,

SOURCE = IUT,

UnconfirmedEventNotification-Request,

Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object detecting the alarm),

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event type),

'Event Type' = (any valid event type),

'Notify Type' = ACK\_NOTIFICATION

9. IF (the notification in step 8 was not broadcast) THEN

IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

RECEIVE

DESTINATION = (at least one device other than the TD),

SOURCE = IUT,

UnconfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object detecting the alarm),

'Time Stamp' = (the timestamp or sequence number from the notification in step 8),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event type),

'Event Type' = (any valid event type),

'Notify Type' = ACK\_NOTIFICATION,

'To State' = (the 'To State' used in step 3 or 4)

ELSE

RECEIVE

DESTINATION = (at least one device other than the TD),

SOURCE = IUT,

UnconfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object detecting the alarm),

'Time Stamp' = (the timestamp or sequence number from the notification in step 8),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event type),

'Event Type' = (any valid event type),

'Notify Type' = ACK\_NOTIFICATION,

10. VERIFY (the 'Event Object Identifier' from the event notification), Acked\_Transitions = (TRUE,TRUE,TRUE)

注意事项：第8步中确认通知使用的目的地址应与第3步中使用的地址相同。步骤9中用于确认通知的目的地址应与步骤4中使用的地址相同。在协议版本1、协议修订版本1中添加了确认通知中包含的“To State”参数。此之前的版本将不包含此参数。当一个特定的事件需要多个事件通知时，IUT传输它们的顺序是不相关的。

1. 使用序号格式的“Time of Acknowledgment”参数时，非确认事件通知成功确认报警

目的：验证是否成功确认由UnconfirmedEventNotification发出的警报，包括通知其他工作站和更新Acked\_Transitions状态。使用序号格式的“Time of Acknowledgment”参数。

测试思想：触发警报使IUT通知TD和至少一个其他设备。TD确认警报，并验证IUT是否正确记录了确认。IUT通知所有其他收件人警报已被确认。

配置要求：IUT应配置至少一个对象，该对象可以检测报警条件并发送未经确认的通知。Acked\_Transitions属性的值应为B'111'，表示所有的转换都已被确认。TD和至少一个其他BACnet设备应是警报通知的接收者。

测试步骤：除确认警报服务请求的“Time of Acknowledgment”参数为序号外，应遵循10.1.1.4中定义的测试步骤。

注意事项：测试通过的结果与10.1.1.4中描述的测试通过消息序列相同。

1. 使用“Time of Acknowledgment”参数的Date Time形式，非确认事件通知成功确认报警

目的：验证是否成功确认由UnconfirmedEventNotification发出的警报，包括通知其他工作站和更新Acked\_Transitions状态。使用Date Time格式的“Time of Acknowledgment”参数。

测试思想：触发警报使IUT通知TD和至少一个其他设备。TD确认警报，并验证IUT是否正确记录了确认。IUT通知所有其他收件人警报已被确认。

配置要求：IUT配置至少一个对象，可以检测报警情况并发送未经确认的通知。Acked\_Transitions属性的值应为B'111'，表示所有的转换都已被确认。TD和至少一个其他BACnet设备应是警报通知的接收者。

测试步骤:应遵循10.1.1.4中定义的测试步骤，但是确认警报服务请求的“Time of Acknowledgment”参数应使用BACnetDateTime格式传递当前时间。

注意事项：测试通过的结果与10.1.1.4中描述的测试通过消息序列相同。

1. 使用“To State” “Offnormal”成功警报确认所有的“Offnormal”转换

目的：确认已成功确认警报，该警报指示“offnormal”状态为“To State”而非“offnormal”。

测试思想:在IUT中触发“offnormal”警报，其中“offnormal”状态由异常以外的事件状态(如高限或低限)表示。TD以异常的“Event State Acknowledged”来确认警报，并验证IUT接受该确认。

配置要求:IUT配置至少一个对象，该对象能够检测报警状态，并且能够进入非正常状态以外的“offnormal”状态。TD应是报警通知的接收方，IUT应配置为未经确认发送。如果IUT不能配置为生成该通知，则应跳过此测试。

测试步骤：

1. MAKE (a change that triggers the detection of an alarm event in the IUT)

2. RECEIVE UnConfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object detecting the alarm),

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (any valid event type),

'Notify Type' = (any valid notify type),

'AckRequired' = TRUE,

'From State' = (any valid event-state),

'To State' = (any "offnormal" event state other than offnormal itself),

'Event Values' = (the values appropriate to the event type)

3. VERIFY(the 'Event Object Identifier' from the event notification), Acked\_Transitions = {0,?,?}

4. TRANSMIT AcknowledgeAlarm-Request,

'Acknowledging Process Identifier' = (the value of the 'Process Identifier' parameter in the event notification),

'Event Object Identifier' = (the 'Event Object Identifier' from the event notification),

'Event State Acknowledged' = offnormal,

'Time Stamp' = (the timestamp conveyed in the notification),

'Time of Acknowledgment' = (any valid timestamp)

5. RECEIVE BACnet-Simple-ACK-PDU

6. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

RECEIVE UnconfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object detecting the alarm),

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (the event type included in step 2),

'Notify Type' = ACK\_NOTIFICATION,

'To State' = (offnormal or the 'To State' from step 2)

ELSE

RECEIVE UnconfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object detecting the alarm),

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (the event type included in step 2),

'Notify Type' = ACK\_NOTIFICATION,

7. VERIFY (the 'Event Object Identifier' from the event notification), Acked\_Transitions = {1,?,?}

1. 使用未知“Acknowledging Process Identifier”参数，确认事件通知成功确认报警

目的：当确认中包含不匹配或未知的“Acknowledging Process Identifier”参数时，验证由ConfirmedEventNotification发出的警报成功确认。

测试思想：触发警报导致IUT通知TD和至少一个其他设备。TD以不匹配的“Acknowledging Process Identifier”(与另一个收件人关联的进程标识符)或未知的“Acknowledging Process Identifier”(与任何收件人无关的进程标识符)确认警报，并确认IUT已正确记录确认。这个测试应该执行两次，一次使用不匹配的进程标识符，一次使用未知的进程标识符。

配置要求：IUT应配置至少一个对象Object1，该对象能够检测报警条件并发送确认的通知。Acked\_Transitions属性应该有一个值(TRUE、TRUE、TRUE)，表示所有的转换都已被确认。TD和至少一个其他BACnet设备应是警报通知的接收者，并应使用不同的进程标识符。

此测试推荐用于所有执行确认警报的BACnet设备，但仅适用于那些说明符合Protocol\_Revision 5或更高版本的设备。

测试步骤：

1. VERIFY (Object1), Acked\_Transitions = (TRUE,TRUE,TRUE)

2. MAKE (a change that triggers the detection of an alarm event in the IUT)

3. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

'Process Identifier' = (any Process ID),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = Object1,

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the Notification Class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (any valid event type),

'Notify Type' = ALARM or EVENT,

'AckRequired' = TRUE,

'From State' = (any appropriate event state),

'To State' = (any appropriate event state),

'Event Values' = (values appropriate to the event type)

4. TRANSMIT BACnet-SimpleACK-PDU

5. RECEIVE

DESTINATION = (at least one device other than the TD),

SOURCE = IUT,

ConfirmedEventNotification-Request,

'Process Identifier' = (any Process ID),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = Object1,

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (any valid event type),

'Notify Type' = ALARM | EVENT,

'AckRequired' = TRUE,

'From State' = (any appropriate event state),

'To State' = (any appropriate event state),

'Event Values' = (values appropriate to the event type)

6. TRANSMIT

DESTINATION = IUT,

SOURCE = (DESTINATION in step 5),

BACnet-SimpleACK-PDU

7. VERIFY (Object1), Acked\_Transitions = (one bit FALSE, the others TRUE)

8. TRANSMIT AcknowledgeAlarm-Request,

'Acknowledging Process Identifier' = (Any mismatched or unknown value),

'Event Object Identifier' = Object1,

'Event State Acknowledged' = (the state specified in the 'To State' parameter of the notification),

'Time Stamp' = (the timestamp conveyed in the notification),

'Time of Acknowledgment' = (the current timestamp)

9. RECEIVE BACnet-SimpleACK-PDU

10. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

'Process Identifier' = (any Process ID),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = Object1,

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the Notification Class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (any valid event type),

'Notify Type' = ACK\_NOTIFICATION,

'To State' = (any appropriate event state)

11. TRANSMIT BACnet-SimpleACK-PDU

12. RECEIVE

DESTINATION = (at least one device other than the TD),

SOURCE = IUT,

ConfirmedEventNotification-Request,

'Process Identifier' = (any Process ID),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = Object1,

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (any valid event type),

'Notify Type' = ACK\_NOTIFICATION,

'To State' = (any appropriate event state)

13. TRANSMIT

DESTINATION = IUT,

SOURCE = (DESTINATION in step 5),

BACnet-SimpleACK-PDU

14. VERIFY (Object1), Acked\_Transitions = (TRUE,TRUE,TRUE)

注意：ConfirmedEventNotification-Request消息可以按任意顺序接收。

1. 使用未知的“Acknowledging Process Identifier”参数，未确认事件通知的成功确认报警

目的：验证当确认中包含不匹配或未知的“Acknowledging Process Identifier”参数时，验证由UnconfirmedEventNotification的警报是否已成功确认。

测试思想：触发警报使IUT通知TD和至少一个其他设备。TD以不匹配的“Acknowledging Process Identifier”(与另一个收件人关联的进程标识符)或未知(与任何收件人都没有关联的进程标识符)确认警报，验证IUT是否正确记录了确认。这个测试应该执行两次，一次使用不匹配的进程标识符，一次使用未知的进程标识符。

配置要求：IUT应配置至少一个对象Object1，该对象可以检测报警条件并发送未经确认的通知。Acked\_Transitions属性应该有一个值(TRUE、TRUE、TRUE)，表示所有的转换都已被确认。TD和至少一个其他BACnet设备应该是警报通知的接收者，配置为接收不同的进程标识符。

此测试推荐用于所有执行确认警报的BACnet设备，但仅适用于那些声明符合Protocol\_Revision 5或更高版本的设备。

测试步骤：

1. VERIFY (Object1), Acked\_Transitions = (TRUE,TRUE,TRUE)

2. MAKE (a change that triggers the detection of an alarm event in the IUT)

3. BEFORE **Notification Fail Time**

RECEIVE UnconfirmedEventNotification-Request,

'Process Identifier' = (any Process ID),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = Object1,

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the Notification Class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (any valid event type),

'Notify Type' = ALARM or EVENT,

'AckRequired' = TRUE,

'From State' = (any appropriate event state),

'To State' = (any appropriate event state),

'Event Values' = (values appropriate to the event type)

4. RECEIVE

DESTINATION = (at least one device other than the TD),

SOURCE = IUT,

UnconfirmedEventNotification-Request,

'Process Identifier' = (any Process ID),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object detecting the alarm),

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (any valid event type),

'Notify Type' = ALARM | EVENT,

'AckRequired' = TRUE,

'From State' = (any appropriate event state),

'To State' = (any appropriate event state),

'Event Values' = (values appropriate to the event type)

5. VERIFY (Object1), Acked\_Transitions = (one bit FALSE, the others TRUE)

6. TRANSMIT AcknowledgeAlarm-Request,

'Acknowledging Process Identifier' = (Any mismatched or unknown value),

'Event Object Identifier' = Object1,

'Event State Acknowledged' = (the state specified in the 'To State' parameter of the notification),

'Time Stamp' = (the timestamp conveyed in the notification),

'Time of Acknowledgment' = (the current timestamp)

7. RECEIVE BACnet-SimpleACK-PDU

8. BEFORE **Notification Fail Time**

RECEIVE UnconfirmedEventNotification-Request,

'Process Identifier' = (any Process ID),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = Object1,

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the Notification Class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (any valid event type),

'Notify Type' = ACK\_NOTIFICATION,

'To State' = (any appropriate event state)

9. RECEIVE

DESTINATION = (at least one device other than the TD),

SOURCE = IUT,

UnconfirmedEventNotification-Request,

'Process Identifier' = (any Process ID),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = Object1,

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (any valid event type),

'Notify Type' = ACK\_NOTIFICATION,

'To State' = (any appropriate event state)

10. VERIFY (Object1), Acked\_Transitions = (TRUE,TRUE,TRUE)

注意：UnconfirmedEventNotification-Request消息可以按任意顺序接收。

1. 确认事件通知的成功报警重新确认

目的：验证由ConfirmedEventNotification发出的的事件是否成功重新确认，包括其他工作站的通知和Acked\_Transitions状态的更新。

测试思想：触发事件并且在事件生成对象的指定Time\_Delay之后，IUT通知TD和至少一个其他设备。TD确认该事件并验证IUT是否正确记录了确认。IUT通知所有收件人事件已被确认。然后TD再次确认事件，IUT再次通知所有接收者。该行为在Protocol\_Revision ≥ 7之前没有定义，因此只有当存在Protocol\_Revision(即Protocol\_Revision ≥ 7)时才执行此测试。

配置要求：IUT应该配置至少一个事件发起对象，该对象生成异常转换并发送确认的通知。Acked\_Transitions属性的值应为B'111'，表示所有的转换都已被确认。TD和至少一个其它BACnet设备应是事件通知的接收者。

测试步骤：

1. MAKE (a change that triggers the detection of an event in the IUT)

2. WAIT (Time\_Delay)

3. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the event-initiating object),

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (any valid event type),

'Notify Type' = (the notify type configured for this event),

'AckRequired' = TRUE,

'From State' = NORMAL,

'To State' = (any appropriate offnormal event state),

'Event Values' = (the values appropriate to the event type)

4. TRANSMIT BACnet-SimpleACK-PDU

5. RECEIVE

DESTINATION = (at least one device other than the TD),

SOURCE = IUT,

ConfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the event-initiating object),

'Time Stamp' = (the timestamp or sequence number received in step 3),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (any valid event type),

'Notify Type' = (the notify type configured for this event),

'AckRequired' = TRUE,

'From State' = NORMAL,

'To State' = (any appropriate offnormal event state),

'Event Values' = (the values appropriate to the event type)

6. TRANSMIT BACnet-SimpleACK-PDU

7. VERIFY (the 'Event Object Identifier' from the event notification), Acked\_Transitions = B'011'

8. TRANSMIT AcknowledgeAlarm-Request,

'Acknowledging Process Identifier' = (the value of the 'Process Identifier' parameter in the event notification),

'Event Object Identifier' = (the 'Event Object Identifier' from the event notification),

'Event State Acknowledged' = (the state specified in the 'To State' parameter of the notification),

'Time Stamp' = (the time stamp conveyed in the notification),

'Acknowledgment Source' = (a character string)

'Time of Acknowledgment' = (any of the forms specified for this parameter)

9. RECEIVE BACnet-Simple-ACK-PDU

10. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the event-initiating object),

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (the event type included in step 3),

'Notify Type' = ACK\_NOTIFICATION,

'To State' = (the 'To State' used in step 3)

11. TRANSMIT BACnet-SimpleACK-PDU

12. RECEIVE

DESTINATION = (at least one device other than the TD),

SOURCE = IUT,

ConfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the event-initiating object),

'Time Stamp' = (the timestamp or sequence number received in step 10),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (the event type included in step 3),

'Notify Type' = ACK\_NOTIFICATION,

'To State' = (the 'To State' used in step 3)

13. TRANSMIT BACnet-SimpleACK-PDU

14. VERIFY (the 'Event Object Identifier' from the event notification), Acked\_Transitions = B'111'

15. TRANSMIT AcknowledgeAlarm-Request,

'Acknowledging Process Identifier' = (the value of the 'Process Identifier' parameter in the event notification),

'Event Object Identifier' = (the 'Event Object Identifier' from the event notification),

'Event State Acknowledged' = (the state specified in the 'To State' parameter of the notification),

'Time Stamp' = (the time stamp conveyed in the notification),

'Acknowledgment Source' = (a character string)

'Time of Acknowledgment' = (any of the forms specified for this parameter)

16. RECEIVE BACnet-SimpleACK-PDU

17. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the event-initiating object),

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (the event type included in step 3),

'Notify Type' = ACK\_NOTIFICATION,

'To State' = (the 'To State' used in step 3)

18. TRANSMIT BACnet-SimpleACK-PDU

19. RECEIVE

DESTINATION = (at least one device other than the TD),

SOURCE = IUT,

ConfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the event-initiating object),

'Time Stamp' = (the timestamp or sequence number received in step 17),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (the event type included in step 3),

'Notify Type' = ACK\_NOTIFICATION,

'To State' = (the 'To State' used in step 3)

20. TRANSMIT BACnet-SimpleACK-PDU

21. VERIFY (the 'Event Object Identifier' from the event notification), Acked\_Transitions = B'111'

注意：步骤12和步骤19中用于确认通知的目标地址应与步骤4中使用的地址相同。当特定事件需要多个事件通知时，IUT传输它们的顺序无关紧要。

1. 非确认事件通知的成功报警重新确认

目的：验证由UnconfirmedEventNotification发出的事件的成功重新确认，包括其他工作站的通知和Acked\_Transitions状态的更新。

测试思想：触发事件并且在事件生成对象指定的Time\_Delay之后，IUT通知TD和至少一个其它设备。TD确认该事件并验证IUT是否正确记录了确认。IUT通知所有接收者事件已被确认。然后TD再次确认事件，IUT再次通知所有接收者。该行为在Protocol\_Revision ≥ 7之前没有定义，因此只有当存在Protocol\_Revision(即Protocol\_Revision ≥ 7)时才执行此测试。

配置要求：IUT应配置至少一个事件发起对象，该对象生成异常转换并发送未经确认的通知。Acked\_Transitions属性的值应为B'111'，表示所有的转换都已被确认。TD和至少一个其他BACnet设备应是事件通知的接收者。

测试步骤：

1. MAKE (a change that triggers the detection of an offnormal event in the IUT)

2. WAIT (Time\_Delay)

3. BEFORE **Notification Fail Time**

RECEIVE UnconfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the event-initiating object),

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (any valid event type),

'Notify Type' = (the notify type configured for this event),

'AckRequired' = TRUE,

'From State' = NORMAL,

'To State' = (any appropriate offnormal event state),

'Event Values' = (the values appropriate to the event type)

4. RECEIVE

DESTINATION = (at least one device other than the TD),

SOURCE = IUT,

UnconfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the event-initiating object),

'Time Stamp' = (the timestamp or sequence number received in step 3),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (any valid event type),

'Notify Type' = (the notify type configured for this event),

'AckRequired' = TRUE,

'From State' = NORMAL,

'To State' = (any appropriate offnormal event state),

'Event Values' = (the values appropriate to the event type)

5. VERIFY (the 'Event Object Identifier' from the event notification), Acked\_Transitions = B'011'

6. TRANSMIT AcknowledgeAlarm-Request,

'Acknowledging Process Identifier' = (the value of the 'Process Identifier' parameter in the event notification),

'Event Object Identifier' = (the 'Event Object Identifier' from the event notification),

'Event State Acknowledged' = (the state specified in the 'To State' parameter of the notification),

'Time Stamp' = (the time stamp conveyed in the notification),

'Acknowledgment Source' = (a character string)

'Time of Acknowledgment' = (any of the forms specified for this parameter)

7. RECEIVE BACnet-SimpleACK-PDU

8. BEFORE **Notification Fail Time**

RECEIVE UnconfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the event-initiating object),

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (the event type included in step 3),

'Notify Type' = ACK\_NOTIFICATION,

'To State' = (the 'To State' used in step 3)

9. RECEIVE

DESTINATION = (at least one device other than the TD),

SOURCE = IUT,

UnconfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the event-initiating object),

'Time Stamp' = (the timestamp or sequence number received in step 8),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (the event type included in step 3),

'Notify Type' = ACK\_NOTIFICATION,

'To State' = (the 'To State' used in step 3)

10. VERIFY (the 'Event Object Identifier' from the event notification), Acked\_Transitions = B'111'

11. TRANSMIT AcknowledgeAlarm-Request,

'Acknowledging Process Identifier' = (the value of the 'Process Identifier' parameter in the event notification),

'Event Object Identifier' = (the 'Event Object Identifier' from the event notification),

'Event State Acknowledged' = (the state specified in the 'To State' parameter of the notification),

'Time Stamp' = (the time stamp conveyed in the notification),

'Acknowledgment Source' = (a character string)

'Time of Acknowledgment' = (any of the forms specified for this parameter)

12. RECEIVE BACnet-SimpleACK-PDU

13. BEFORE **Notification Fail Time**

RECEIVE UnconfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the event-initiating object),

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (the event type included in step 3),

'Notify Type' = ACK\_NOTIFICATION,

'To State' = (the 'To State' used in step 3)

14. RECEIVE

DESTINATION = (at least one device other than the TD),

SOURCE = IUT,

UnconfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the event-initiating object),

'Time Stamp' = (the timestamp or sequence number received in step 13),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event),

'Event Type' = (the event type included in step 3),

'Notify Type' = ACK\_NOTIFICATION,

'To State' = (the 'To State' used in step 3)

15. VERIFY (the 'Event Object Identifier' from the event notification), Acked\_Transitions = B'111'

注意：步骤9和步骤14中用于确认通知的目的地址应与步骤4中使用的地址相同。当特定事件需要多个事件通知时，IUT传输它们的顺序无关紧要。

1. AcknowledgeAlarm服务负响应执行测试

此测试组的目的是在预期服务失败的情况下验证AcknowledgeAlarm服务请求的正确执行。所有测试用例都代表了AcknowledgeAlarm服务参数可能与对象的当前警报状态不一致的方式。

1. 由于“Time Stamp”太旧导致确认事件通知的确认报警不成功

目的：如果确认中的时间戳与最近一次转换到当前警报状态的时间戳不匹配，验证报警仍然未被确认。

测试思想：触发警报并导致IUT通知TD和至少一个其他设备。TD使用旧的时间戳来确认警报，并验证该确认未被IUT接受，且IUT没有通知其他设备该警报已被确认。然后，TD使用正确的时间戳确认警报，并确认IUT已正确记录确认。IUT通知所有其他收件人警报已被确认。

配置要求：配置IUT使至少一个对象，可以检测报警情况并发送确认的通知。Acked\_Transitions属性的值应为B'111'，表示所有的转换都已被确认。TD和至少一个其他BACnet设备应是警报通知的接收者。

测试步骤：

1. MAKE (a change that triggers the detection of an alarm event in the IUT)
2. WAIT (Time\_Delay)
3. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object detecting the alarm),

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event type),

'Event Type' = (any valid event type),

'Notify Type' = (the notify type configured for the event),

'AckRequired' = TRUE,

'From State' = NORMAL,

'To State' = (any appropriate non-normal event state),

'Event Values' = (the values appropriate to the event type)

1. TRANSMITBACnet-SimpleACK-PDU
2. RECEIVE

DESTINATION = (at least one device other than the TD),

SOURCE = IUT,

ConfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object detecting the alarm),

'Time Stamp' = (the timestamp or sequence number received in step 3),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event type),

'Event Type' = (any valid event type),

'Notify Type' = (the notify type configured for the event),

'AckRequired' = TRUE,

'From State' = NORMAL,

'To State' = (any appropriate non-normal event state),

'Event Values' = (the values appropriate to the event type)

1. TRANSMITBACnet-SimpleACK-PDU
2. VERIFY (the 'Event Object Identifier' from the event notification), Acked\_Transitions =

(FALSE,TRUE,TRUE)

1. TRANSMIT AcknowledgeAlarm-Request,

'Acknowledging Process Identifier' = (the value of the 'Process Identifier' parameter in the event notification),

'Event Object Identifier' = (the 'Event Object Identifier' from the event notification),

'Event State Acknowledged' = (the state specified in the 'To State' parameter of the notification),

'Time Stamp' = (a time stamp older than the one conveyed in the notification),

'Time of Acknowledgment' = (the current time using a Time format)

1. RECEIVEBACnet-Error-PDU,

Error Class = SERVICES,

Error Code = INVALID\_TIME\_STAMP

1. VERIFY (the 'Event Object Identifier' from the event notification), Acked\_Transitions = (FALSE,TRUE,TRUE)
2. TRANSMITAcknowledgeAlarm-Request,

'Acknowledging Process Identifier' = (the process identifier configured for this event),

'Event Object Identifier' = (the 'Event Object Identifier' from the event notification),

'Event State Acknowledged' = (the state specified in the 'To State' parameter of the notification),

'Time Stamp' = (the time stamp conveyed in the notification),

'Time of Acknowledgment' = (the current time using a Time format)

1. RECEIVE BACnet-Simple-ACK-PDU
2. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

BEFORE **Notification Fail Time**

RECEIVE

ConfirmedEvent Notification-Request**,**

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object detecting the alarm),

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event type),

'Event Type' = (any valid event type),

'Notify Type' = ACK\_NOTIFICATION,

'To State' = (the 'To State' used in step 3 or 5)

ELSE

BEFORE **Notification Fail Time**

RECEIVE

ConfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object detecting the alarm),

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event type),

'Event Type' = (any valid event type),

'Notify Type' = ACK\_NOTIFICATION

1. TRANSMIT BACnet-SimpleACK-PDU
2. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1)THEN

RECEIVE

ConfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object detecting the alarm),

'Time Stamp' = (the timestamp or sequence number from the notification in step 13),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event type),

'Event Type' = (any valid event type),

'Notify Type' = ACK\_NOTIFICATION,

'To State' = (the 'To State' used in step 3 or 5)

ELSE

RECEIVE

DESTINATION = (at least one device other than the TD),

SOURCE = IUT,

ConfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object detecting the alarm),

'Time Stamp' = (the timestamp or sequence number from the notification in step 13),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event type),

'Event Type' = (any valid event type),

'Notify Type' = ACK\_NOTIFICATION

1. TRANSMIT BACnet-SimpleACK-PDU
2. VERIFY (the 'Event Object Identifier' from the event notification), Acked\_Transitions = (TRUE,TRUE,TRUE)

注意事项：步骤11中确认通知使用的目标地址应与步骤3中使用的地址相同。

1. 由于“Acknowledging Process Identitifier”无效导致确认事件通知的确认警报不成功

该测试已被移除。

1. 由于引用对象不存在导致确认事件通知的确认警报不成功

目标：验证如果“Event Object Identifier”表示的对象不存在时，则警报仍然未被确认。

测试思想：触发警报且导致IUT通知TD与至少一个的其他设备。TD使用不正确的“Event Object Identifier”来识别警报，并检验IUT不接受该确认，且IUT不会通知其他设备该警报信息已被确认。然后TD使用适当的“Event Object Identifier”来识别警报信息，并验证IUT是否正确地记录了确定信息。IUT通知所有其他接收者已经确认警报。

配置要求：配置IUT使至少一个对象，可以检测报警情况并发送确认通知。Acked\_Transitions值应为B'111'，表示所有的转换都已被确认。TD和至少一个其他BACnet设备应是警报通知的接收者。

测试步骤：应当遵循10.1.2.1中定义的测试步骤，但在第一个AcknowledgeAlarm中，要求“Time Stamp”和事件通知里的“Time Stamp”相同，“Event Object Identifier”的值应与事件通知中的“Event Object Identifier”不同，并且IUT对象不存在。

注意事项：传递的结果与10.1.2.1中描述的消息序列相同，只是步骤7中的错误类为OBJECT，步骤7中的错误代码为UNKNOWN\_OBJECT。对于Protocol\_Revision为5或更早版本的设备，还应接受错误代码为INCONSISTENT\_PARAMETERS的错误服务类。

1. 由于“Event State Acknowledged”无效导致确认事件通知的确认警报不成功

目的：验证如果“Event State Acknowledged”与定义正在确认的警报的其他参数不一致，则警报仍然不能被确认。

测试思想：触发警报并导致IUT通知TD与至少一个的其他设备。TD使用不正确的“Event State Acknowledged”来识别警报，并检验IUT不接受该确认，且IUT不会通知其他设备该警报信息已被确认。然后TD使用正确的“Event State Acknowledged”来识别警报信息，并验证IUT是否正确地记录了确定信息。IUT通知所有其他接收者已经确认警报。

配置要求：IUT至少配置一个对象，可以检测报警情况并发送确认通知。Acked\_Transitions值应为B'111'，表示所有的转换都已被确认。TD和至少一个其他BACnet设备应是警报通知的接收者。

测试步骤：应遵循10.1.2.1中定义的测试步骤，但在第一个AcknowledgeAlarm请求中，“Time Stamp”应与事件通知中的“Time Stamp”相同，且“Event State Acknowledged”应具有非正常值，而不是OFFNORMAL，也不是事件通知中“To State”参数的值。

注意事项:传递的结果与10.1.2.1中描述的消息序列相同，只是步骤7中的错误代码应为INVALID\_EVENT\_STATE。对于Protocol\_Revision为5或更早版本的设备，还应接受错误代码为INCONSISTENT\_PARAMETERS的错误服务类。

1. 由于“Time Stamp”太旧导致非确认事件通知的不成功确认报警

目的：如果确认中的时间戳与最近一次转换到当前警报状态的时间戳不匹配，则验证报警仍然未被确认。

测试思想：触发警报，IUT通知TD和至少一个其他设备。TD使用旧的时间戳来确认警报，并验证该确认未被IUT接受，且IUT没有通知其它设备该警报已被确认。然后，TD会使用正确的时间戳确认警报，并确认IUT已正确记录确认。IUT通知所有其它接收者已经确认警报。

配置要求：IUT至少配置一个对象，可以检测报警情况并发送确认通知。Acked\_Transitions值应为B'111'，表示所有的转换都已被确认。TD和至少一个其他BACnet设备应是警报通知的接收者。

1. MAKE (a change that triggers the detection of an alarm event in the IUT)
2. WAIT (Time\_Delay)
3. BEFORE **Notification Fail Time**

RECEIVE UnconfirmedEventNotification-Request

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT

'Event Object Identifier' = (the object detecting the alarm),

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event type),

'Event Type' = (any valid event type),

'Notify Type' = (the notify type configured for the event),

'AckRequired' = TRUE,

'From State' = NORMAL,

'To State' = (any appropriate non-normal event state),

'Event Values' = (the values appropriate to the event type)

1. IF (the notification in step 3 was not a broadcast) THEN

RECEIVE

DESTINATION = (at least one device other than the TD),

SOURCE = IUT,

UnconfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object detecting the alarm),

'Time Stamp' = (the timestamp or sequence number from step 3),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event type),

'Event Type' = (any valid event type),

'Notify Type' = (the notify type configured for the event),

'AckRequired' = TRUE,

'From State' = NORMAL,

'To State' = (any appropriate non-normal event state),

'Event Values' = (the values appropriate to the event type)

1. VERIFY (the 'Event Object Identifier' from the event notification), Acked\_Transitions =

(FALSE,TRUE,TRUE)

1. TRANSMIT AcknowledgeAlarm-Request,

'Acknowledging Process Identifier' = (the value of the 'Process Identifier' parameter in the

event notification),

'Event Object Identifier' = (the 'Event Object Identifier' from the event notification),

'Event State Acknowledged' = (the state specified in the 'To State' parameter of the

notification),

'Time Stamp' = (a time stamp older than the one conveyed in the notification),

'Time of Acknowledgment' = (the TD’s current time using a Time format)

1. RECEIVE BACnet-Error-PDU,

Error Class = SERVICES,

Error Code = INVALID\_TIME\_STAMP

1. VERIFY (the 'Event Object Identifier' from the event notification), Acked\_Transitions =

(FALSE,TRUE,TRUE)

1. TRANSMIT AcknowledgeAlarm-Request,

'Acknowledging Process Identifier' = (the process identifier configured for this event),

'Event Object Identifier' = (the 'Event Object Identifier' from the event notification),

'Event State Acknowledged' = (the state specified in the 'To State' parameter of the

notification),

'Time Stamp' = (the time stamp conveyed in the notification),

'Time of Acknowledgment' = (the TD’s current time using a Time format)

1. RECEIVE BACnet-Simple-ACK-PDU
2. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

BEFORE **Notification Fail Time**

RECEIVE

DESTINATION = LOCAL BROADCAST | GLOBAL BROADCAST | TD,

SOURCE = IUT,

UnconfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object detecting the alarm),

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event type),

'Event Type' = (any valid event type),

'Notify Type' = ACK\_NOTIFICATION,

'To State' = (the 'To State' used in step 3 or 4)

ELSE

BEFORE **Notification Fail Time**

RECEIVE

DESTINATION = LOCAL BROADCAST | GLOBAL BROADCAST | TD,

SOURCE = IUT,

UnconfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' =IUT,

'Event Object Identifier' = (the object detecting the alarm),

'Time Stamp' = (the current time or sequence number),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event type),

'Event Type' = (any valid event type),

'Notify Type' = ACK\_NOTIFICATION

1. IF (the notification in step 11 was not broadcast) THEN

IF (Protocol\_Revision is present and Protocol\_Revision ≥ 1) THEN

RECEIVE

DESTINATION = (at least one device other than the TD),

SOURCE = IUT,

UnconfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object detecting the alarm),

'Time Stamp' = (the timestamp or sequence number from the notification in step 11),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event type),

'Event Type' = (any valid event type),

'Notify Type' = ACK\_NOTIFICATION,

'To State' = (the 'To State' used in step 3 or 4)

ELSE

RECEIVE

DESTINATION = (at least one device other than the TD),

SOURCE = IUT,

UnconfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),

'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the object detecting the alarm),

'Time Stamp' = (the timestamp or sequence number from the notification in step 11),

'Notification Class' = (the notification class configured for this event),

'Priority' = (the priority configured for this event type),

'Event Type' = (any valid event type),

'Notify Type' = ACK\_NOTIFICATION

1. VERIFY (the 'Event Object Identifier' from the event notification), Acked\_Transitions =

(TRUE,TRUE,TRUE)

注意事项：步骤11中确认通知使用的目标地址应与步骤3中使用的地址相同。步骤12中用于确认通知的目标地址应与步骤4中使用的地址相同。当当特定事件需要多个事件通知时，IUT传输它们的顺序无关紧要。

1. 由于引用对象不存在导致非确认事件通知的确认警报不成功

目的：验证如果“Event Object Identifier”表示的对象不存在时，则警报未被确认。

测试思想：触发警报且导致IUT通知TD和其他至少一个其他设备。TD使用无效的Event Object Identifier来识别警报，并验证IUT不接受该确认，且IUT不通知其他设备该警报已被确认。然后TD使用适当的事件对象标识符来识别警报，并验证IUT正确地记录了确认。IUT通知所有其他接收者警报已被确认。

配置要求：配置IUT使少一个对象，可以检测警报情况并发送未经确认的通知。Acked\_Transitions属性的值为B'111'，表示所有的转换都已被确认。TD和至少一个其他BACnet设备应是警报通知的接收者。

测试步骤：应遵循10.1.2.5中定义的测试步骤，但在第一个确认警报请求中，“Time Stamp”的值应与事件通知中的“Time Stamp”相同，“Event Object Identifier”的值应与事件通知中的“Event Object Identifier”不同，且IUT中不存在对象。

注意事项：传递结果与10.1.2.5中传递结果描述的消息序列相同，但是步骤7中的错误类应该是OBJECT，步骤7中的错误代码应该是UNKNOWN\_OBJECT。对于Protocol\_Revision为5或更早版本的设备，还应接受错误代码为INCONSISTENT\_PARAMETERS的错误服务等级。

1. 由于“Event State Acknowledged”无效导致非确认事件通知的确认警报不成功

目的：验证如果“Event State Acknowledged”与定义正在确认的警报的其他参数不一致，则警报不能被确认。

测试思想：触发警报，IUT通知TD和至少一个其他设备。TD使用一个无效的“Event State Acknowledged”来确认该警报，验证该确认未被IUT接受，且IUT没有通知其他设备该警报已被确认。然后TD使用正确的“Event State Acknowledged”来确认警报，并验证IUT是否正确记录了确认。IUT通知所有其他接收者警报已被确认。

配置要求：配置IUT使得至少一个对象，可以检测报警情况并发送未经确认的通知。Acked\_Transitions的值为B'111'，表示所有的转换都已被确认。TD和至少一个其他BACnet设备应是警报通知的接收者。

测试步骤：应遵循10.1.2.1中定义的测试步骤，但在第一个AcknowledgeAlarm请求中，“Time Stamp”应与事件通知中的“Time Stamp”相同，且“Event State Acknowledged”应具有非正常值，而不是OFFNORMAL，也不是事件通知中“To State”参数的值。

注意事项：传递结果与10.1.2.5中传递结果描述的消息序列相同，但是步骤7中的错误类应该是OBJECT，步骤7中的错误代码应该是UNKNOWN\_OBJECT。对于Protocol\_Revision为5或更早版本的设备，还应接受错误代码为INCONSISTENT\_PARAMETERS的错误服务等级。

* 1. 确认COV通知（ConfirmedCOVNotification）服务执行测试

这章节定义了证明支持执行ConfirmedCOVNotification服务请求所必需的测试。ConfirmedCOVNotification测试适用于提供内部COV报告功能的特定对象类型。IUT应通过每个标准BACnet对象类型的所有测试，这些测试具有可选或必需的内在COV报告功能。

所需测试：SubscribeCOV服务初始化测试9.10。

BACnet参考条款:13.6。

1. ConfirmedCOVNotification服务正响应执行测试

此测试组的目的是在预期服务能够成功完成的情况下，验证ConfirmedCOVNotification服务请求的正确执行。

1. 模拟、二进制、多状态和生命安全对象的值更改通知

COV通知在模拟输入、模拟输出、模拟值、二进制输入、二进制输出、二进制值、多状态输入、多状态输出、多状态值、生命安全点和生命安全区对象启动时传递Present\_Value和Status\_Flags属性的值。

由于订阅COV通知的功能是通用的，并且可以应用于这些对象类型中的任何一种，因此IUT必须证明其正确响应了来自代表这些对象类型中每个对象的COV通知。本节所定义的测试程序应适用于任何一种对象类型。

目的：验证IUT可以执行来自模拟、二进制和多状态对象的ConfirmedCOVNotification请求。

测试步骤：

REPEAT X = (one object of each type in the set {Analog Input, Analog Output, Analog Value, Binary Input, Binary Output, Binary Value, Multi-state Input, Multi-state Output, Multi-state Value, Life Safety Point, Life Safety Zone})

DO {

1. RECEIVE SubscribeCOV,

'Subscriber Process Identifier' = (any valid process identifier),

'Monitored Object Identifier' = X,

'Issue Confirmed Notifications ' = TRUE,

'Lifetime' = (a value greater than one minute)

1. TRANSMIT BACnet-SimpleACK-PDU
2. TRANSMIT ConfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the process identifier used in step 2),

'Initiating Device Identifier' = TD,

'Monitored Object Identifier' = X,

'Time Remaining' = (the time remaining in the subscription),

'List of Values' = (Present\_Value and Status\_Flags appropriate to object X)

1. RECEIVE BACnet-SimpleACK-PDU
2. CHECK (to ensure that any appropriate functions defined by the manufacturer, such as displaying information on a workstation screen are carried out)

}

1. 循环（Loop）对象值更改通知

目的：验证IUT可以执行来自循环对象的ConfirmedCOVNotification请求。

测试步骤：

1. RECEIVE SubscribeCOV,

'Subscriber Process Identifier' = (any valid process identifier),

'Monitored Object Identifier' = (any Loop object, X),

'Issue Confirmed Notifications ' = TRUE,

'Lifetime' = (a value greater than one minute)

1. TRANSMIT BACnet-SimpleACK-PDU
2. TRANSMIT ConfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the process identifier used in step 2),

'Initiating Device Identifier' = TD,

'Monitored Object Identifier' = X,

'Time Remaining' = (the time remaining in the subscription),

'List of Values' = (Present\_Value, Status\_Flags, Setpoint, and Controlled\_Variable\_Value appropriate to object X)

1. RECEIVE BACnet-SimpleACK-PDU
2. CHECK (to ensure that any appropriate functions defined by the manufacturer, such as displaying

information on a workstation screen are carried out)

1. ConfirmedCOVNotification服务负响应执行测试

此测试组的目的是验证在预期服务失败的情况下，ConfirmedCOVNotification服务请求的正确执行。从IUT的角度来看，所有测试用例均表示ConfirmedCOVNotification服务参数可能与预订的当前状态不一致的方式。

1. 订阅过期后更改值到达通知

目的：在订阅时间段到期后，如果COV通知到达，验证是否返回了相应的错误。

测试步骤：

1. RECEIVE SubscribeCOV,

'Subscriber Process Identifier' = (any valid process identifier),

'Monitored Object Identifier' = (any object X of a type that supports COV notification),

'Issue Confirmed Notifications' = TRUE,

'Lifetime' = (a value no greater than one minute)

1. TRANSMIT BACnet-SimpleACK-PDU
2. WAIT (a value two times Lifetime)
3. TRANSMIT ConfirmedCOVNotification-Request

'Subscriber Process Identifier' = (the process identifier used in step 2),

'Initiating Device Identifier' = TD,

'Monitored Object Identifier' = X,

'Time Remaining' = (any amount of time greater than 0),

'List of Values' = (a list of values appropriate to object X)

1. RECEIVE BACnet-Error-PDU,

Error Class = SERVICES,

Error Code = (any valid error code for class SERVICES)

1. 无效进程标识符的值更改通知

目的：如果到达的COV通知包含与任何与当前订阅不匹配的进程标识符，验证是否返回了相应的错误。

测试步骤：

1. RECEIVE SubscribeCOV,

'Subscriber Process Identifier' = (any valid process identifier),

'Monitored Object Identifier' = (any object X of a type that supports COV notification),

'Issue Confirmed Notifications' = TRUE,

'Lifetime' = (a value no greater than one minute)

1. TRANSMIT BACnet-SimpleACK-PDU
2. TRANSMIT ConfirmedCOVNotification-Request,

'Subscriber Process Identifier' =(a process identifier different from the one used in step 2),

'Initiating Device Identifier' = TD,

'Monitored Object Identifier' = X,

'Time Remaining' = (any amount of time greater than 0),

'List of Values' = (a list of values appropriate to object X)

1. RECEIVE BACnet-Error-PDU,

Error Class = SERVICES,

Error Code = (any valid error code for class SERVICES)

1. 使用无效启动设备标识符的值更改通知

目的：如果COV通知到达且包含与当前订阅不匹配的启动设备标识符，则验证是否返回了相应的错误。

测试步骤：

1. RECEIVE SubscribeCOV,

'Subscriber Process Identifier' = (any valid process identifier),

'Monitored Object Identifier' = (any object X of a type that supports COV notification),

'Issue Confirmed Notifications' = TRUE,

'Lifetime' = (a value no greater than one minute)

1. TRANSMIT BACnet-SimpleACK-PDU
2. TRANSMIT ConfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the process identifier different used in step 2),

'Initiating Device Identifier' = (any valid Device object except TD),

'Monitored Object Identifier' = X,

'Time Remaining' = (any amount of time greater than 0),

'List of Values' = (a list of values appropriate to object X)

1. RECEIVE BACnet-Error-PDU,

Error Class = SERVICES,

Error Code = (any valid error code for class SERVICES)

1. 使用无效受监视对象标识符的值更改通知

目的：如果到达的COV通知包含任何与当前订阅不匹配的受监视对象标识符，验证是否返回了相应的错误。

测试步骤：

1. RECEIVE SubscribeCOV,

'Subscriber Process Identifier' = (any valid process identifier),

'Monitored Object Identifier' = (any object X of a type that supports COV notification),

'Issue Confirmed Notifications' = TRUE,

'Lifetime' = (a value no greater than one minute)

1. TRANSMIT BACnet-SimpleACK-PDU
2. TRANSMIT ConfirmedCOVNotification-Request,

'Subscriber Process Identifier' = the process identifier used in step 2),

'Initiating Device Identifier' = TD,

'Monitored Object Identifier' = (any object Y supporting COV notification except X),

'Time Remaining' = (any amount of time greater than 0),

'List of Values' = (a list of values appropriate to object Y)

1. RECEIVE BACnet-Error-PDU,

Error Class = SERVICES,

Error Code = (any valid error code for class SERVICES)

1. 使用无效的值列表的值更改通知

目的：目的：如果到达的COV通知包含不适用于受监视对象的对象类型的值列表，则验证是否返回了相应的拒绝。

测试步骤：

1. RECEIVE SubscribeCOV,

'Subscriber Process Identifier' = (any valid process identifier),

'Monitored Object Identifier' = (any object X of a type that supports COV notification),

'Issue Confirmed Notifications' = TRUE,

'Lifetime' = (a value no greater than one minute)

1. TRANSMIT BACnet-SimpleACK-PDU
2. TRANSMIT ConfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the process identifier used in step 2),

'Initiating Device Identifier' = TD,

'Monitored Object Identifier' = X,

'Time Remaining' = (any amount of time greater than 0),

'List of Values' = (a list of values that is not appropriate to object X)

1. RECEIVE BACnet-Reject-PDU,

'Reject Reason' = INCONSISTENT\_PARAMETERS |

INVALID\_PARAMETER\_DATATYPE| INVALID\_TAG

* 1. 未确认COV通知（UnconfirmedCOVNotification）服务执行测试

1. 设备重启通知

目的：验证IUT执行重启通知传递的UnconfirmedCOVNotification服务请求。

测试思想：TD发送重启通知，测试人员验证IUT是否以供应商指定的方式处理通知。

测试配置：应选择一个有效的BACnetDeviceStatus和一个有效的Last\_Restart\_Reason对，DS1和LRR1，IUT将对其采取某些措施。。

设备重启通知与使用UnconfirmedCOVNotification服务的订阅COV通知在两个方面有所不同。首先，通过Restart\_Notification\_Recipients属性而不是SubscribeCOV进行订阅。其次，“Subscriber Process Identifier”参数的值始终为零。

测试步骤：

1. TRANSMIT UnconfirmedCOVNotification-Request,

'Subscriber Process Identifier' = 0,

'Initiating Device Identifier' = TD,

'Monitored Object Identifier' = (the TD Device Identifier),

'Time Remaining' = 0,

'List of Values' = (System\_Status =DS1,

Time\_Of\_Device\_Restart =(any valid time),

Last\_Restart\_Reason = LRR1)

1. CHECK(that the IUT processes the notification as described by the vendor)
   1. 确认事件通知（ConfirmedEventNotification）服务执行测试

所需测试：没有。

BACnet参考条款:13.8。

目的：验证IUT可以执行ConfirmedEventNotification服务请求。

测试思想：除返回确认外BACnet不定义在收到ConfirmedEventNotification之后要采取的任何操作。虽然返回确认信息就足以符合标准，但供应商可以指定能够观察到的其他操作。任何供应商定义的在测试期间未发生的操作都应在测试报告中注明。不包括负响应试验。

1. 使用Time格式的“Timestamp”参数并传递文本消息的ConfirmedEventNotification

测试步骤：

1. TRANSMIT ConfirmedEventNotification-Request,

'Process Identifier' = (any valid process identifier),

'Initiating Device Identifier' = TD,

'Event Object Identifier' = (any Event Enrollment object),

'Time Stamp' = (current time using the Time format),

'Notification Class' = (any valid notification class),

'Priority' = (any valid priority),

'Event Type' = (any standard event type),

'Message Text' = (any character string),

'Notify Type' = ALARM | EVENT,

'AckRequired' = FALSE,

'From State' = NORMAL,

'To State' = (any non-normal state appropriate to the event type),

'Event Values' = (any values appropriate to the event type)

1. RECEIVE BACnet-SimpleACK-PDU
2. CHECK (for any vendor-defined observable actions)
3. 使用Date Time格式的“Timestamp”参数且没有文本消息的ConfirmedEventNotification

测试步骤：

1. TRANSMIT ConfirmedEventNotification-Request,

'Process Identifier' = (any valid process identifier),

'Initiating Device Identifier' = TD,

'Event Object Identifier' = (any Event Enrollment object),

'Time Stamp' = (current time using the DateTime format),

'Notification Class' = (any valid notification class),

'Priority' = (any valid priority),

'Event Type' = (any standard event type),

'Notify Type' = ALARM | EVENT,

'AckRequired' = FALSE,

'From State' = NORMAL,

'To State' = (any non-normal state appropriate to the event type),

'Event Values' = (any values appropriate to the event type)

1. RECEIVE BACnet-SimpleACK-PDU
2. CHECK (for any vendor-defined observable actions)
3. 使用序号格式的“Timestamp”参数且没有文本消息的ConfirmedEventNotification

测试步骤：

1. TRANSMIT ConfirmedEventNotification-Request,

'Process Identifier' = (any valid process identifier),

'Initiating Device Identifier' = TD,

'Event Object Identifier' = (any Event Enrollment object),

'Time Stamp' = (current sequence number),

'Notification Class' = (any valid notification class),

'Priority' = (any valid priority),

'Event Type' = any standard event type),

'Notify Type' = ALARM | EVENT,

'AckRequired' = FALSE,

'From State' = NORMAL,

'To State' = (any non-normal state appropriate to the event type),

'Event Values' = (any values appropriate to the event type)

1. RECEIVE BACnet-SimpleACK-PDU
2. CHECK (for any vendor-defined observable actions)
3. 不带通知类参数的ConfirmedEventNotification

测试已移除。

1. ConfirmedEventNotification的简单显示

目的：这个测试用例验证IUT能够最低限度地显示ConfirmedEventNotification。

配置：对于此测试，测试人员应该选择一个事件生成对象，O1。

测试步骤：

1. TRANSMIT ConfirmedEventNotification-Request,

'Process Identifier' = (a valid process identifier specified by the IUT vendor),

'Initiating Device Identifier' = TD,

'Event Object Identifier' = O1,

'Time Stamp' = (current time in any format),

'Notification Class' = (any valid notification class),

'Priority' = (any valid priority),

'Event Type' = (any standard event type),

'Message Text' = (any character string),

'Notify Type' = ALARM | EVENT,

'AckRequired' = TRUE | FALSE,

'From State' = (state S1, any valid state for this event type),

'To State' = (state S2, any valid state for this event type that can follow S1),

'Event Values' = (any values appropriate to the event type)

1. RECEIVE BACnet-SimpleACK-PDU
2. CHECK (that the IUT indicates the notification to the operator and that the indication includes identification of the event generating object or the monitored object, the event timestamp, and the event Message Text)
3. CHECK (that all information indicated to the user is consistent with the information provided in step 1)

传递结果：如果消息文本超过IUT可显示的最大长度，IUT将截断该消息文本。允许IUT在显示的文本中包含字符，表明消息已被截断，即使截断的消息短于32个字符。IUT不能截断长度小于或等于32个字符的消息文本。

1. ConfirmedEventNotification的完整显示

目的：这个测试用例用于验证IUT是否能够显示ConfirmedEventNotifications。

配置：对于此测试，测试人员应该选择一个事件生成对象，O1。

测试步骤：

1. TRANSMIT ConfirmedEventNotification-Request,

'Process Identifier' = (a valid process identifier specified by the IUT vendor),

'Initiating Device Identifier' = TD,

'Event Object Identifier' = O1,

'Time Stamp' = (current time in any format),

'Notification Class' = (any valid notification class),

'Priority' = (any valid priority),

'Event Type' = (any standard event type),

'Message Text' = (any character string),

'Notify Type' = ALARM | EVENT,

'AckRequired' = TRUE | FALSE,

'From State' = (state S1, any valid state for this event type),

'To State' = (state S2, any valid state for this event type that can follow S2),

'Event Values' = (any values appropriate to the event type)

1. RECEIVE BACnet-SimpleACK-PDU
2. CHECK (that the IUT indicates the notification to the operator and that the indication includes identification of the event generating object or the monitored object, the event timestamp, the event Message Text, Notification Class, Priority, Notify Type, Ack Required, To State and Event Values)
3. CHECK (that all information indicated to the user is consistent with the information provided in step 1)

传递结果：如果消息文本长于IUT可显示的最大长度，则IUT将截断消息文本。允许IUT在显示的文本中包含表示消息已被截断的字符，即使截短的消息长度之后会小于255个字符。 IUT不得截断长度小于或等于255个字符的消息文本。

* 1. 未确认事件通知（UnconfirmedEventNotification）服务执行测试

1. UnconfirmedEventNotification简单显示

该测试与第10.4.5节ConfirmedEventNotification简单演示中的测试相同，除了在步骤1中已经发送TRANSMITUnconfirmedEventNotification，并且在步骤2中IUT没有返回BACnet-SimpleACK-PDU。

1. UnconfirmedEventNotification完整显示

该测试与第10.4.6节ConfirmedEventNotification简单显示中的测试相同，除了在步骤1中已经发送TRANSMITUnconfirmedEventNotification，并且在步骤2中IUT没有返回BACnet-SimpleACK-PDU。

* 1. 获取警报摘要（GetAlarmSummary）服务执行测试

本节定义了证明执行GetAlarmSummary服务请求所必需的测试。

BACnet参考条款：13.10。

所需测试：无。

1. 没有活动警报的警报摘要

目的：验证IUT在没有可报告的活动警报时可以执行GetAlarmSummary服务请求。

配置要求：应配置IUT，使其不存在活动警报。

测试步骤：

1.TRANSMIT GetAlarmSummary-Request

2.RECEIVEGetAlarmSummary-ACK，

'List of Alarm Summaries' = (an empty list)

1. 一个活动警报的警报摘要

目的：当只有一个可报告的活动警报时，验证IUT是否可以执行GetAlarmSummary服务请求。

配置要求：IUT应配置为只有一个活动警报。

测试步骤：

1.TRANSMIT GetAlarmSummary-Request

2.RECEIVE GetAlarmSummary-ACK，

'List of Alarm Summaries' = (a list with exactly one entry corresponding to the known active alarm)

1. 具有多个活动警报的警报摘要

目的：验证当有多个可报告的活动警报时，IUT可以执行GetAlarmSummary服务请求。 此测试用例仅对包含多个可检测警报的对象的设备执行。

配置要求：应配置IUT，以便有多个活动警报。

测试步骤：

1.TRANSMIT GetAlarmSummary-Request

2.RECEIVEGetAlarmSummary-ACK，

'List of Alarm Summaries' = (a list containing one entry for each known active alarm)

* 1. 获取是事件摘要（GetEnrollmentSummary）服务执行测试

本节定义了证明支持执行GetEnrollmentSummary服务请求所必需的测试。

所需测试：无。

BACnet参考条款：13.11。

测试思想：此服务有6个过滤器，可用于选择事件注册。在6个过滤器中，只需要“Acknowledgment Filter”。对于每个测试用例，需要特定的配置来产生所用过滤器的预期结果。10.7.1中的测试用例仅使用所需的“Acknowledgment Filter”。因此，应为每个测试用例配置IUT。10.7.2中的测试用例使用可选的过滤器，目的是为IUT配置丰富的事件注册数据库，这些数据库具有适合这些过滤器的功能，但某些情况下实施起来是不可能的。不必支持提供与每个过滤器条件匹配的事件注册的配置。但是要求可以执行所有测试并根据IUT数据库的配置返回正确的响应。如果无法用满足过滤条件的方法来配置IUT，那么应该返回空的注册列表。此测试没有负响应测试。

1. 必需的GetEnrollmentSummary过滤器

目的：此测试组用于在预期服务成功完成且仅使用所需的“Acknowledgment Filter”的情况下验证GetEnrollmentSummary服务请求的正确执行。

1. 零摘要的注册摘要

目的：在没有要报告的注册时验证IUT是否可以执行GetEnrollmentSummary请求。

配置要求：IUT应配置为无需报告的注册。

测试步骤：

1. TRANSMIT GetEnrollmentSummary-Request,

'Acknowledgment Filter' = ALL

2. RECEIVE GetEnrollmentSummary-ACK,

'List of Enrollment Summaries' = (an empty list)

1. ACKED

目的：验证在Acknowledgement Filter已经设置为ACKED时IUT可以执行GetEnrollmentSummary请求。

配置要求：配置IUT至少两个事件注册，其中一个事件用于所有事件转换已被确认，另一个用于至少还有一个未确认的事件转换。

测试步骤：

1. TRANSMIT GetEnrollmentSummary-Request,

'Acknowledgment Filter' = ACKED

2. RECEIVE GetEnrollmentSummary-ACK,

'List of Enrollment Summaries' = (all events for which the event transitions have all been acknowledged)

1. NOT-ACKED

目的：验证在Acknowledgement Filter设置为NOT-ACKED时IUT可以执行GetEnrollmentSummary请求。

配置要求：IUT应配置至少两个事件注册，其中一个事件用于所有事件转换已被确认，另一个用于至少还有一个未确认的事件转换。

测试步骤：

1. TRANSMIT GetEnrollmentSummary-Request,

'Acknowledgment Filter' = NOT-ACKED

2. RECEIVE GetEnrollmentSummary-ACK,

'List of Enrollment Summaries' = (all events for which there is at least one unacknowledged event transition)

1. All

目的：验证IUT在Acknowledgement Filter的过滤器请求已经设置为ALL时，是否可以执行GetEnrollmentSummary请求。

配置要求：IUT应配置至少两个事件注册，其中一个事件用于所有事件转换已被确认，另一个用于至少还有一个未确认的事件转换。

测试步骤：

1. TRANSMIT GetEnrollmentSummary-Request,

'Acknowledgment Filter' = ALL

2. RECEIVE GetEnrollmentSummary-ACK,

'List of Enrollment Summaries' = (the union of the summaries provided in 10.7.1.2 and 10.7.1.3)

1. 用户可选的GetEnrollmentSummary过滤器

目的：此测试组用于验证在预期服务成功完成并使用用户可选过滤器的情况下GetEnrollmentSummary服务请求的正确执行。

1. 注册过滤器

目的：验证在使用“Enrollment Filter”时IUT可以执行GetEnrollmentSummary。

配置要求：如果可能配置IUT，以便至少有两个注册摘要报告不同的（BACnetRecipient，Process Identifier）对。 TD将在GetEnrollmentSummary服务请求中使用这些组合之一。

测试步骤：

1. TRANSMIT GetEnrollmentSummary-Request,

'Acknowledgment Filter' = ALL,

'Enrollment Filter' = (one of the (BACnetRecipient, Process Identifier) pairs configured for this test)

1. RECEIVE GetEnrollmentSummary-ACK,

'List of Enrollment Summaries' = (all enrollments configured with this (BACnetRecipient, Process Identifier) pair)

1. 事件状态过滤器

目的：验证在使用“Event State Filter”时IUT可以执行GetEnrollmentSummary请求。

配置要求：如果可能配置IUT，使其具有一个或多个事件生成对象，其Event\_State属性值为NORMAL，一个或多个具有属性值为FAULT的Event\_State，一个或多个具有属性值 OFFNORMAL的Event\_State，以及一个或多个属性值不是NORMAL，OFFNORMAL或FAULT的Event\_State。如果只支持这些情况的一部分，则应配置尽可能多的情况。

测试步骤：

1. TRANSMIT GetEnrollmentSummary-Request,

'Acknowledgment Filter' = ALL,

'Event State Filter' = NORMAL

1. RECEIVE GetEnrollmentSummary-ACK,

'List of Enrollment Summaries' = (all configured event-generating objects with Event\_State =

NORMAL)

1. TRANSMIT GetEnrollmentSummary-Request,

'Acknowledgment Filter' = ALL,

'Event State Filter' = FAULT

1. RECEIVE GetEnrollmentSummary-ACK,

'List of Enrollment Summaries' = (all configured event-generating objects with Event\_State =

FAULT)

1. TRANSMIT GetEnrollmentSummary-Request,

'Acknowledgment Filter' = ALL,

'Event State Filter' = OFFNORMAL

1. RECEIVE GetEnrollmentSummary-ACK,

'List of Enrollment Summaries' = (all configured event-generating objects with Event\_State =

OFFNORMAL)

1. TRANSMIT GetEnrollmentSummary-Request,

'Acknowledgment Filter' = ALL,

'Event State Filter' = ACTIVE

1. RECEIVE GetEnrollmentSummary-ACK,

'List of Enrollment Summaries' = (all configured event-generating objects with Event\_State =

a value other than NORMAL)

1. TRANSMIT GetEnrollmentSummary-Request,

'Acknowledgment Filter' = ALL,

'Event State Filter' = ALL

1. RECEIVE GetEnrollmentSummary-ACK,

'List of Enrollment Summaries' =(the union of all of the summaries returned in steps 1 - 8)

1. 事件类型过滤器

目的：验证在使用“Event Type Filter”时IUT是否可以执行GetEnrollmentSummary请求。

配置要求：如果可能配置IUT，使其具有一个或多个事件生成对象，用于以下每个事件类型CHANGE\_OF\_BITSTRING，CHANGE\_OF\_STATE，CHANGE\_OF\_VALUE，COMMAND\_FAILURE，FLOATING\_LIMIT和OUT\_OF\_RANGE。如果仅支持这些事件类型的子集，应该尽可能多地配置它们。

测试步骤:

1. TRANSMIT GetEnrollmentSummary-Request,

'Acknowledgment Filter' = ALL,

'Event Type Filter' = CHANGE\_OF\_BITSTRING

1. RECEIVE GetEnrollmentSummary-ACK,

'List of Enrollment Summaries' = (all configured event-generating objects with

Event\_Type = CHANGE\_OF\_BITSTRING)

1. TRANSMIT GetEnrollmentSummary-Request,

'Acknowledgment Filter' = ALL,

'Event Type Filter' = CHANGE\_OF\_STATE

1. RECEIVE GetEnrollmentSummary-ACK,

'List of Enrollment Summaries' = (all configured event-generating objects with

Event\_Type = CHANGE\_OF\_STATE)

1. TRANSMIT GetEnrollmentSummary-Request,

'Acknowledgment Filter' = ALL,

'Event Type Filter' = CHANGE\_OF\_VALUE

1. RECEIVE GetEnrollmentSummary-ACK,

'List of Enrollment Summaries' = (all configured event-generating objects with

Event\_Type = CHANGE\_OF\_VALUE)

1. TRANSMIT GetEnrollmentSummary-Request,

'Acknowledgment Filter' = ALL,

'Event Type Filter' = FLOATING\_LIMIT

1. RECEIVE GetEnrollmentSummary-ACK,

'List of Enrollment Summaries' = (all configured event-generating objects with

Event\_Type =FLOATING\_LIMIT)

1. 优先级过滤器

目的：验证IUT在使用“Priority Filter”时可以执行GetEnrollmentSummary请求。

配置要求：如果可能，IUT应在四个不同的优先级中配置一个或多个事件生成对象。为了达到此测试目的，事件生成对象的优先级是对象最近转换的优先级。优先级应为0，Xlow，Xhigh，255，其中10 <Xlow<100且100 <Xhigh<255。如果一次只能支持这些优先级的子集，则应配置尽可能多的优先级。

测试步骤：

1. TRANSMIT GetEnrollmentSummary-Request,

'Acknowledgment Filter' = ALL,

'MinPriority ' = 0,

'MaxPriority' = 0

1. RECEIVE GetEnrollmentSummary-ACK,

'List of Enrollment Summaries' = (all configured event-generating objects with a priority in the specified range)

1. TRANSMIT GetEnrollmentSummary-Request,

'Acknowledgment Filter' = ALL,

'MinPriority' = 0,

'MaxPriority' = 255

1. RECEIVE GetEnrollmentSummary-ACK,

'List of Enrollment Summaries' = (all configured event-generating objects with a priority in the specified range)

1. TRANSMIT GetEnrollmentSummary-Request,

'Acknowledgment Filter' = ALL,

'MinPriority' = Xlow,

'MaxPriority' = Xhigh

1. RECEIVE GetEnrollmentSummary-ACK,

'List of Enrollment Summaries' = (all configured event-generating objects with a priority in the specified range)

1. 通知类过滤器

目的：验证IUT在使用“Notification Class Filter”时可以执行GetEnrollmentSummary请求。

配置要求：如果可能，IUT应使用两个通知类中的每一个配置一个或多个事件生成对象。

测试步骤：

1. TRANSMIT GetEnrollmentSummary-Request,

'Acknowledgment Filter' = ALL,

'Notification Class Filter ' = (any of the configured notification classes)

1. RECEIVE GetEnrollmentSummary-ACK,

'List of Enrollment Summaries' = (all configured event-generating objects using the specified notification class)

1. 过滤器组合

目的：验证在使用用户可选过滤器组合时IUT可以执行GetEnrollmentSummary请求。

配置要求：10.7.2.1-10.7.2.5中定义的事件生成对象配置的任意组合都可以的。

测试步骤：

1. TRANSMIT GetEnrollmentSummary-Request,

'Acknowledgment Filter' = ALL | ACKED | NOT\_ACKED,

(any combination of user selectable filters chosen by the TD with appropriate values),

1. RECEIVE GetEnrollmentSummary-ACK,

'List of Enrollment Summaries' = (all configured event-generating objects matching all of the filter requirements)

* 1. 获取事件信息（GetEventInformation）服务执行测试

本节定义了演示支持执行GetEventInformation服务请求所必需的测试。

所需测试：无。

BACnet参考条款：13.12。

1. 没有活动事件的事件信息

目的：在没有要报告的活动事件时验证IUT可以执行GetEventInformation服务请求。

配置要求：应配置IUT，使其不存在活动事件状态。

测试步骤：

1. TRANSMIT GetEventInformation-Request
2. RECEIVE GetEventInformation-ACK,

'List of Event Summaries' = (an empty list),

'More Events' = FALSE

1. 包含一个活动事件的事件信息

目的：验证当只有一个可报告的活动事件时，IUT可以执行GetEventInformation服务请求。

配置要求：配置IUT只有一个活动事件状态。

测试步骤：

1. TRANSMIT GetEventInformation-Request
2. RECEIVE GetEventInformation-ACK,

'List of Event Summaries' = (a list with exactly one entry corresponding to the known active event),

'More Events' = FALSE

1. 具有多个活动事件的事件信息

目的：当有多个活动事件状态要报告时，验证IUT可以执行GetEventInformation服务请求。此测试用例仅对包含多个可检测警报的对象的设备执行。

配置要求：IUT应配置为有多个活动事件状态，但要求传输的次数少于'More Events'= TRUE时的传输次数。如果IUT不能配置成包含多个可以在没有“More Events”= TRUE时的单个GetEventInformation服务响应中传输的活动事件，则应跳过此测试。

测试步骤：

1. TRANSMITGetEventInformation-Request
2. RECEIVE GetEventInformation-ACK,

'List of Event Summaries' = (a list containing one entry for each known active event state),

'More Events' = FALSE

1. 基于Event\_State的事件信息

目的：验证IUT可以在Event\_State属性引起活动事件状态时执行GetEventInformation服务请求，而不是由未确认的事件转换引起。

配置要求：IUT应配置为只有一个活动事件状态，该状态是由属性不是NORMAL的Event\_State引起的，并且Acknowledged\_Transitions属性的所有位都设置为TRUE。

测试步骤：

1. TRANSMITGetEventInformation-Request
2. RECEIVEGetEventInformation-ACK，

'List of Event Summaries' = (a list with exactly one entry corresponding to the known active event state),

'More Events'= FALSE

1. 基于Acknowledged\_Transitions的事件信息

目的：当活动事件状态由未确认的事件转换而不是Event\_State属性引起时，验证IUT可以执行GetEventInformation服务请求。

配置要求：IUT应配置为只有一个活动事件状态，并且该状态是由属性为NORMAL的Event\_State，以及属性至少有一个位设置为FALSE的Acknowledged\_Transitions引起的。

测试步骤：

1. TRANSMITGetEventInformation-Request
2. RECEIVEGetEventInformation-ACK，

'List of Event Summaries' = (a list with exactly one entry corresponding to the known active event state),

'More Events'= FALSE

1. 链式测试

目的：此测试用例使用多个GetEventInformation消息来测试链接功能。

配置要求：IUT应配置为使得事件状态可在多于128字节的单个APDU中传送的事件状态。 IUT应配置成包含可以触发链接效果的足够的事件。如果IUT不能配置为包含足够的活动事件来触发链接，则应跳过此测试。

测试思想：在步骤1-4中，测试首先通过从IUT请求两个列表并验证第二个列表与第一个列表是否不同来测试正确的链接。在步骤5-9中，为了测试BACnet 13.12.1.1.1中定义的“固定对象处理顺序”，它再次请求第一个列表，然后，在请求第二个列表之前，使第一个列表中生成的最后一个对象列表不再具有任何活动事件状态。当TD使用当前正常设备的对象标识符请求第二列表时，IUT应该响应与之前相同的第二列表。

测试步骤:

1. TRANSMIT GetEventInformation-Request,

'max-APDU-length-accepted' = B'0000',

'segmented-response-accepted' = FALSE

1. RECEIVE GetEventInformation-ACK,

'List of Event Summaries' = (an arbitrary list),

'More Events' = TRUE

1. TRANSMIT GetEventInformation-Request,

'Last Received Object Identifier' = the last object identifier of the list received in step2)

1. RECEIVE GetEventInformation-ACK,

'List of Event Summaries' = (a list of object identifiers not including any received in step 2)

1. TRANSMIT GetEventInformation-Request,

'max-APDU-length-accepted' = B'0000',

'segmented-response-accepted' = FALSE

1. RECEIVE GetEventInformation-ACK,

'List of Event Summaries' = (an arbitrary list),

'More Events' = TRUE

1. MAKE (the object identified by the last object identifier in the list received in step 6 have no active

event states)

1. TRANSMIT GetEventInformation-Request,

'Last Received Object Identifier' = (the last object identifier of the list received in step6)

1. RECEIVE GetEventInformation-ACK,

'List of Event Summaries' = (the same list received in step 4)

* 1. 生命安全操作（LifeSafetyOperation）服务执行测试

本节定义了证明支持执行LifeSafetyOperation服务请求所必需的测试。

1. 重置单个对象执行测试

所需测试：无。

BACnet参考条款：13.13。

目的：验证IUT是否可以正确执行LifeSafetyOperation服务请求到一个Life Safety Object。

测试思想：Life Safety Object在NORMAL和OFFNORMAL状态之间切换。STATUS应锁定在OFFNORMAL状态直到LifeSafetyOperation服务请求TRANSMIT完毕。当设备不支持锁存时，可以省略该测试。

配置要求：IUT必须支持可以警报的Life Safety Object。此对象配置为NORMAL状态，驱动为OFFNORMAL状态，并通过调整其Present\_Value返回NORMAL状态。STATUS应保持锁定在OFFNORMAL状态直到复位。必须对支持的每个复位功能重复此测试。

测试步骤:

1. REPEAT X = (All supported enumerations that reset the object) DO {
2. MAKE (the selected object enter a latched state where enumeration X will reset the object)
3. MAKE (Event-State = NORMAL)
4. IF (enumeration value creates an Event\_State = FAULT) THEN

CHECK (Event\_State = FAULT)

ELSE

CHECK (Event\_State = OFFNORMAL)

1. TRANSMIT LifeSafetyOperation-Request,

'Requesting Process Identifier' = (any valid identifier),

'Requesting Source' = (any valid character string),

'Request' = (any valid LifeSafetyOperation request),

'Object Identifier' = (the selected object)

1. RECEIVE BACnet-SimpleACK-PDU
2. VERIFY (Object), STATUS = NORMAL)

}

1. 重置多个对象执行测试

所需测试：无。

BACnet参考条款：13.13。

目的：验证IUT是否可以正确地对多个Life Safety Object执行LifeSafetyOperation服务请求。

测试思想：两个Life Safety Object在NORMAL和OFFNORMAL状态之间切换。STATUS应锁定在OFFNORMAL状态直到LifeSafetyOperation服务请求传输完毕。当设备不支持锁存时，可以省略该测试。

配置要求：IUT必须支持可以警报的Life Safety Object，并且至少有两个可以锁存的对象。此对象配置为NORMAL状态，驱动为OFFNORMAL状态，并通过调整其Present\_Value返回NORMAL状态。STATUS应保持锁定在OFFNORMAL状态直到复位。对每个支持的复位功能重复此测试。

测试步骤:

1. REPEAT X = (All supported enumerations that reset the object) DO {
2. MAKE (the selected objects enter a latched state where enumeration X will reset the objects)
3. MAKE (Event\_State = NORMAL)
4. IF (enumeration value creates an Event\_State = FAULT) THEN

CHECK (Event\_State = FAULT)

ELSE

CHECK (Event\_State = OFFNORMAL)

1. TRANSMIT LifeSafetyOperation-Request,

'Requesting Process Identifier' = (any valid identifier),

'Requesting Source' = (any valid character string),

'Request' = (any valid LifeSafetyOperation request)

1. RECEIVE BACnet-SimpleACK-PDU
2. VERIFY (Object), STATUS = NORMAL)

}

1. 静音执行测试

所需测试：无。

BACnet参考条款：13.13。

目的：验证IUT是否可以正确执行LifeSafetyOperation服务请求使警报设备静音。

测试思想：声音设备和/或视觉设备连接到IUT并且正在发声/闪烁。发送一个LifeSafetyOperation服务请求使发声器/频闪器静止。

配置要求：IUT必须配备所需的声光设备。

测试步骤：

1. REPEAT X = (All supported enumerations that silence the object) DO {
2. MAKE (the selected object enter a state where enumeration X will commence alerts)
3. MAKE (Event\_State = NORMAL)
4. TRANSMIT LifeSafetyOperation-Request,

'Requesting Process Identifier' = (any valid identifier),

'Requesting Source' = (any valid character string),

'Request' = (any valid LifeSafetyOperation request),

'Object Identifier' = (the selected object)

1. RECEIVE BACnet-SimpleACK-PDU
2. CHECK (Sounder/Strobe inactive)

}

* 1. 订阅COV（SubscribeCOV）服务执行测试

本节定义了证明支持执行SubscribeCOV服务请求所必需的测试。

所需测试：无。

BACnet参考条款：13.14。

配置要求：IUT应配置至少一个支持COV通知订阅的对象。

1. 正响应SubscribeCOV服务执行测试

此测试组的目的是在预期服务成功完成的情况下验证SubscribeCOV服务请求的正确执行。

1. 确认COV通知

目的：验证IUT是否正确响应SubscribeCOV请求以建立已确认的COV通知的订阅。支持COV报告的实现无法响应此测试和10.10.1.2中的测试的错误。

测试步骤:

1. TRANSMIT SubscribeCOV-Request,

'Subscriber Process Identifier' = (any valid process identifier),

'Monitored Object Identifier' = (any object supporting COV notifications),

'Issue Confirmed Notifications' = TRUE,

'Lifetime' = (any value > 0 if automatic cancellation is supported, otherwise 0)

1. RECEIVE BACnet-SimpleACK-PDU
2. WAIT **Notification Fail Time**
3. IF (the IUT supports confirmed notifications) THEN

RECEIVE ConfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = (any value > 0 if automatic cancellation is supported, otherwise 0),

'List of Values' = (values appropriate to the object type of the monitored object)

ELSE

RECEIVE BACnet-Error PDU,

Error Class = SERVICES,

Error Code = SERVICE\_REQUEST\_DENIED | OTHER

1. 未确认COV通知

目的：验证IUT是否正确响应了SubscribeCOV请求以建立未确认的COV通知的订阅。支持COV报告的实现无法响应此测试和10.10.1.1中的测试的错误。

测试步骤:

1. TRANSMIT SubscribeCOV-Request,

'Subscriber Process Identifier' = (any valid process identifier),

'Monitored Object Identifier' = (any object supporting COV notifications),

'Issue Confirmed Notifications' = FALSE,

'Lifetime' = (any value > 0 if automatic cancellation is supported, otherwise 0)

1. RECEIVE BACnet-SimpleACK-PDU
2. WAIT **Notification Fail Time**
3. IF (the IUT supports confirmed notifications) THEN

RECEIVE UnconfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = (any value > 0 if automatic cancellation is supported, otherwise 0),

'List of Values' = (values appropriate to the object type of the monitored object)

ELSE

RECEIVE BACnet-Error PDU,

Error Class = SERVICES,

Error Code = SERVICE\_REQUEST\_DENIED | OTHER

1. 显式无限生命期COV订阅

目的：验证IUT是否正确响应了SubscribeCOV请求以建立具有无限生命期期（lifetime = 0）的订阅。可以使用已确认或未确认的通知，但IUT必须至少支持其中一个选项。

测试步骤：

1. TRANSMIT SubscribeCOV-Request,

'Subscriber Process Identifier' = (any valid process identifier),

'Monitored Object Identifier' = (any object supporting COV notifications),

'Issue Confirmed Notifications' = TRUE | FALSE,

'Lifetime' = 0

1. RECEIVE BACnet-SimpleACK-PDU
2. WAIT **Notification Fail Time**
3. IF (the subscription was for confirmed notifications) THEN,

RECEIVE ConfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = 0,

'List of Values' = (values appropriate to the object type of the

monitored object)

ELSE

RECEIVE UnconfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = 0,

'List of Values' = (values appropriate to the object type of the

monitored object)

1. MAKE (a change to the monitored object that should cause a COV notification)
2. IF (the subscription was for confirmed notifications) THEN

RECEIVE ConfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = 0,

'List of Values' = (values appropriate to the object type of the

monitored object including the changed value that

triggered the notification)

ELSE

RECEIVE UnconfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = 0,

'List of Values' = (values appropriate to the object type of the

monitored object including the changed value of

that triggered the notification)

1. 取消COV 订阅

所需测试：无限生命期COV订阅，10.10.1.1。

目的：验证IUT是否正确响应SubscribeCOV请求以取消COV订阅。此测试取消了10.10.1.1中的订阅。

测试步骤：

1. TRANSMIT SubscribeCOV-Request，

'Subscriber Process Identifier' = (the process identifier used in test 9.10.1.1),

'Monitored Object Identifier' = (the same object used in test 9.10.1.1)

1. RECEIVE BACnet-SimpleACK-PDU
2. WAIT **Notification Fail Time**
3. MAKE (a change to the monitored object that would cause a COV notification if there were an active

subscription)

1. CHECK (verify that the IUT did not transmit a COV notification message)
2. 取消过期或不存在的订阅

目的：验证IUT是否正确响应SubscribeCOV请求以取消不再存在的订阅。

测试步骤：

1. TRANSMIT SubscribeCOV-Request,

'Subscriber Process Identifier' = (any unused process identifier or an identifier from

a previously terminated subscription),

'Monitored Object Identifier' = (any unused object or an object from a previously

terminated subscription)

1. RECEIVE BACnet-SimpleACK-PDU
2. WAIT **Notification Fail Time**
3. MAKE (a change to the monitored object that would cause a COV notification if there were an active

subscription)

1. CHECK (verify that the IUT did not transmit a COV notification message)
2. 隐含无限生命期COV订阅

目的：验证IUT是否正确响应了SubscribeCOV请求以建立具有隐含的无限生命期的订阅（省略了生命期参数）。可以使用已确认或未确认的通知，但IUT必须至少支持其中一个选项。

测试步骤：测试步骤与10.10.1.1相同，但省略步骤1中的“Lifetime”参数。

测试人员注意事项：传递结果与10.10.1.1中的结果相同。

1. 有限生命期订阅

目的：验证IUT是否正确响应SubscribeCOV请求以建立具有临时生命期的订阅。 可以使用已确认或未确认的通知，但IUT必须至少支持其中一个选项。

1. TRANSMIT SubscribeCOV-Request,

'Subscriber Process Identifier' = (any valid process identifier),

'Monitored Object Identifier' = (any object supporting COV notifications),

'Issue Confirmed Notifications' = TRUE | FALSE,

'Lifetime' = (a value between 60 seconds and 300 seconds)

1. RECEIVE BACnet-SimpleACK-PDU
2. WAIT **Notification Fail Time**
3. IF (the subscription was for confirmed notifications) THEN

RECEIVE ConfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = (the requested subscription lifetime),

'List of Values' = (values appropriate to the object type of the

monitored object)

ELSE

RECEIVE UnconfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = (the requested subscription lifetime),

'List of Values' = (values appropriate to the object type of the

monitored object)

1. MAKE (a change to the monitored object that should cause a COV notification)
2. IF (the subscription was for confirmed notifications) THEN

RECEIVE ConfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = (a value greater than 0 and less than the requested

subscription lifetime),

'List of Values' = (values appropriate to the object type of the

monitored object)

ELSE

RECEIVE UnconfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = (a value greater than 0 and less than the requested

subscription lifetime),

'List of Values' = (values appropriate to the object type of the

monitored object including the changed value of that triggered the notification)

1. WAIT (the lifetime of the subscription)
2. MAKE (a change to the monitored object that would cause a COV notification if there were an active

subscription)

1. CHECK (verify that the IUT did not transmit a COV notification message)
2. 更新现有订阅

目的：验证IUT是否正确响应SubscribeCOV请求，以更新订阅的生命周期。可以使用已确认或未确认的通知，但IUT必须至少支持其中一个选项。

测试思想：COV通知的订阅持续60秒。在该订阅到期之前，第二次订阅将持续300秒。当响应第二个订阅发送通知时，检查生命周期以验证它是否大于60但小于300秒。

1. TRANSMIT SubscribeCOV-Request,

'Subscriber Process Identifier' = (any valid process identifier),

'Monitored Object Identifier' = (any object supporting COV notifications),

'Issue Confirmed Notifications' = TRUE | FALSE,

'Lifetime' = 60

1. RECEIVE BACnet-SimpleACK-PDU
2. WAIT **Notification Fail Time**
3. IF (the subscription was for confirmed notifications) THEN

RECEIVE ConfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = 60,

'List of Values' = (values appropriate to the object type of the

monitored object)

ELSE

RECEIVE UnconfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = 60,

'List of Values' = (values appropriate to the object type of the

monitored object)

1. TRANSMIT SubscribeCOV-Request,

'Subscriber Process Identifier' = (any valid process identifier),

'Monitored Object Identifier' = (any object supporting COV notifications),

'Issue Confirmed Notifications' = TRUE | FALSE,

'Lifetime' = (a value between 180 and 300 seconds)

1. RECEIVE BACnet-SimpleACK-PDU
2. WAIT **Notification Fail Time**
3. IF (the subscription was for confirmed notifications)THEN

RECEIVE ConfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = (the requested subscription lifetime),

'List of Values' = (values appropriate to the object type of the

monitored object)

ELSE

RECEIVE UnconfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = (the requested subscription lifetime),

'List of Values' = (values appropriate to the object type of the

monitored object)

1. 确保订阅生命周期不受时间变化的影响

所需测试：TimeSynchronization服务执行测试，10.30; UTCTimeSynchronization服务执行测试，10.31。

目的：验证IUT是否正确响应了SubscribeCOV请求以建立具有临时生命期的订阅，并且生命期不受TimeSynchronization或UTC-TimeSynchronization服务请求的影响。

测试思想：TD订阅了支持IUT中COV报告的对象。时间更改为D1，它在将来或过去会超过300秒。更改对象，生成COV通知并验证是否发出了COV通知。然后允许订阅到期并且再次改变对象，如果存在订阅则将生成COV通知。验证没有生成通知。

测试配置：IUT包含支持COV报告的对象。如果IUT不支持TimeSynchronization或UTC-TimeSynchronization，则应省略此测试。

测试步骤：

1. TRANSMIT SubscribeCOV-Request,

'Subscriber Process Identifier' = (any valid process identifier),

'Monitored Object Identifier' = (any object supporting COV notifications),

'Issue Confirmed Notifications' = TRUE | FALSE,

'Lifetime' = (a value between 60 seconds and 300 seconds)

1. RECEIVE BACnet-SimpleACK-PDU
2. BEFORE **Notification Fail Time**

IF (the subscription was for confirmed notifications) THEN

RECEIVE ConfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = (the requested subscription lifetime),

'List of Values' = (values appropriate to the object type of the

monitored object)

TRANSMIT BACnet-SimpleACK-PDU

ELSE

RECEIVE UnconfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = (the requested subscription lifetime),

'List of Values' = (values appropriate to the object type of the

monitored object)

1. MAKE (a change to the monitored object that should cause a COV notification)
2. BEFORE **NotificationFailTime**

IF(the subscription was for confirmed notifications)THEN

RECEIVE ConfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = (a value greater than 0 and less than the requested subscription lifetime),

'List of Values' = (values appropriate to the object type of the monitored object)

TRANSMIT BACnet-SimpleACK-PDU

ELSE

RECEIVE UnconfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = (a value greater than 0 and less than the

requested subscription lifetime),

'List of Values' = (values appropriate to the object type of the

monitored object including the changed value of that triggered the notification)

1. TRANSMIT

DA = GLOBAL BROADCAST,

SA = TD,

BACnet-Unconfirmed-Request-PDU,

'Service Choice' = TimeSynchronization-Request,

date = D1,

time = D1

1. TRANSMIT

DA = GLOBAL BROADCAST,

SA = TD,

BACnet-Unconfirmed-Request-PDU,

'Service Choice' = UTC-TimeSynchronization-Request,

date = D1,

time = D1

1. MAKE (a change to the monitored object that should cause a COV notification)
2. BEFORE **NotificationFailTime**

IF(the subscription was for confirmed notifications) THEN

RECEIVE ConfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = (a value greater than 0 and less than the

requested subscription lifetime),

'List of Values' = (values appropriate to the object type of the

monitored object)

TRANSMIT BACnet-SimpleACK-PDU

ELSE

RECEIVE UnconfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription)

'Initiating Device Identifier' = IUT,

Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = (a value greater than 0 and less than the

requested subscription lifetime),

'List of Values' = (values appropriate to the object type of the

monitored object including the changed

value of that triggered the notification)

1. WAIT (the remaining lifetime of the subscription)
2. MAKE (a change to the monitored object that would cause a COV notification if there were an active

subscription)

1. CHECK (verify that the IUT did not transmit a COV notification message)
2. 负响应SubscribeCOV服务执行测试

此测试组的目的是在预期服务失败的情况下验证SubscribeCOV服务请求的正确执行。

1. 受监视对象不支持COV通知

目的：验证当受监视对象不支持COV通知时，IUT正确响应SubscribeCOV请求以建立订阅。

测试步骤:

1. TRANSMIT SubscribeCOV-Request,

'Subscriber Process Identifier' = (any valid process identifier),

'Monitored Object Identifier' = (any object that does not support COV notifications),

'Issue Confirmed Notifications' = TRUE,

'Lifetime' = 60

1. RECEIVE BACnet-Error PDU,

Error Class = SERVICES,

Error Code = SERVICE\_REQUEST\_DENIED | OTHER

* 1. 订阅COV属性（SubscribeCOVProperty） 服务执行测试

本节定义了证明支持执行SubscribeCOVProperty服务请求所必需的测试。

所需测试：无。

BACnet参考条款：13.15。

配置要求：IUT应配置至少一个支持COV通知订阅的对象。

1. 正响应SubscribeCOVProperty服务执行测试

此测试组的目的是在预期服务成功完成的情况下验证SubscribeCOVProperty服务请求的正确执行。

1. 确认COV通知

目的：验证IUT是否正确响应SubscribeCOVProperty请求以建立已确认COV通知的订阅。支持COV报告的实现无法响应此测试和10.11.1.2中的测试的错误。

测试步骤：

1. TRANSMIT SubscribeCOVProperty-Request,

'Subscriber Process Identifier' = (any valid process identifier),

'Monitored Object Identifier' = (any object supporting COV notifications),

'Issue Confirmed Notifications' = TRUE,

'Lifetime' = (any value > 0 if automatic cancellation is supported, otherwise 0),

'Monitored Property Identifier' = (any valid property supporting COV notifications)

1. RECEIVE BACnet-SimpleACK-PDU
2. BEFORE **Notification Fail Time**

IF (the IUT supports confirmed notifications) THEN

RECEIVE BACnetConfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = (any value > 0 if automatic cancellation is supported,otherwise 0),

'List of Values' = (values appropriate to the property subscribed to,

and any other properties the IUT provides with it,

such as Status-Flags)

ELSE

RECEIVE BACnet-Error PDU,

Error Class = SERVICES,

Error Code = SERVICE\_REQUEST\_DENIED | OTHER

1. 未确认COV通知

目的：验证IUT是否正确响应SubscribeCOVProperty请求以建立未确认COV通知的订阅。支持COV报告的实现无法响应此测试和10.11.1.1中的测试的错误。

测试步骤：

1. TRANSMIT SubscribeCOVProperty-Request,

'Subscriber Process Identifier' = (any valid process identifier),

'Monitored Object Identifier' = (any object supporting COV notifications),

'Issue Confirmed Notifications' = TRUE,

'Lifetime' = (any value > 0 if automatic cancellation is supported, otherwise 0),

'Monitored Property Identifier' = (any valid property supporting COV notifications)

1. RECEIVE BACnet-SimpleACK-PDU
2. BEFORE **Notification Fail Time**

IF (the IUT supports confirmed notifications) THEN

RECEIVE BACnetUnconfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = (any value > 0 if automatic cancellation is supported,

otherwise 0),

'List of Values' = (values appropriate to the property subscribed to,

and any other properties the IUT provides with it, such as Status-Flags)

ELSE

RECEIVE BACnet-Error PDU,

Error Class = SERVICES,

Error Code = SERVICE\_REQUEST\_DENIED | OTHER

1. 明确无限生命期COV订阅

目的：验证IUT是否正确响应SubscribeCOVProperty请求以建立具有无限生命期（life = 0）的订阅。可以使用已确认或未确认的通知，但IUT必须至少支持其中一个选项。

测试步骤：

1. TRANSMIT SubscribeCOVProperty-Request,

'Subscriber Process Identifier' = (any valid process identifier),

'Monitored Object Identifier' = (any object supporting COV notifications),

'Issue Confirmed Notifications' = TRUE | FALSE,

'Lifetime' = 0,

'Monitored Property Identifier' = (any valid property supporting COV notifications)

1. RECEIVE BACnet-SimpleACK-PDU
2. BEFORE **Notification Fail Time**

IF (the subscription was for confirmed notifications) THEN

RECEIVE BACnetConfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = 0,

'List of Values' = (values appropriate to the property subscribed to,

and any other properties the IUT provides with it,

such as Status\_Flags)

TRANSMIT BACnet-SimpleACK-PDU

ELSE

RECEIVE BACnetUnconfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = 0,

'List of Values' = (values appropriate to the property subscribed to,

and any other properties the IUT provides with it,

such as Status\_Flags)

1. MAKE (a change to the monitored object that should cause a COV notification)
2. BEFORE **Notification Fail Time**

IF (the subscription was for confirmed notifications) THEN

RECEIVE BACnetConfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = 0,

'List of Values' = (values appropriate to the property subscribed to

including the changed value that triggered the

notification, and any other properties the IUT

provides with it, such as Status\_Flags)

ELSE

RECEIVE BACnetUnconfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = 0,

'List of Values' = (values appropriate to the property subscribed to

including the changed value that triggered the

notification, and any other properties the IUT

provides with it, such as Status-Flags)

1. 取消COV订阅

所需测试：无限生命期COV订阅, 10.11.1.1。

目的：验证IUT是否正确响应SubscribeCOVProperty请求以取消COV订阅。此测试取消了10.11.1.1中的订阅。

测试步骤:

1. TRANSMIT SubscribeCOVProperty-Request,

'Subscriber Process Identifier' = (the process identifier used in test 10.11.1.1),

'Monitored Object Identifier' = (the same object used in test 10.11.1.1),

'Monitored Property Identifier' = (the same property used in test 10.11.1.1)

1. RECEIVE BACnet-SimpleACK-PDU
2. WAIT **Notification Fail Time**
3. MAKE (a change to the monitored object that would cause a COV notification if there were an active

subscription)

注意事项：IUT不得传输COV通知消息。

1. 取消过期或不存在的订阅

目的：验证IUT是否正确响应SubscribeCOVProperty请求以取消不再存在的订阅。

测试步骤:

1. TRANSMIT SubscribeCOVProperty-Request,

'Subscriber Process Identifier' = (any unused process identifier or an identifier from

a previously terminated subscription),

'Monitored Object Identifier' = (any unused object or an object from a previously

terminated subscription),

'Monitored Property Identifier' = (any unused property or a property from a

previously terminated subscription)

1. RECEIVE BACnet-SimpleACK-PDU
2. WAIT **Notification Fail Time**
3. MAKE (a change to the monitored object that would cause a COV notification if there were an active

subscription)

注意事项：IUT不得传输COV通知消息。错误消息不是可接受的响应。

1. 隐含的无限生命期COV订阅

目的：验证IUT是否正确响应SubscribeCOVProperty请求以建立具有隐含的无限生命期的订阅（省略了生命期参数）。可以使用已确认或未确认的通知，但IUT必须至少支持其中一个选项。

测试步骤：除了省略步骤1中的“Lifetime”参数外，测试步骤与10.11.1.1相同。

注意事项：传递结果与10.11.1.1中的结果相同。

1. 有限生命期订阅

目的：验证IUT是否正确响应SubscribeCOVProperty请求以建立具有临时生命期的订阅。可以使用已确认或未确认的通知，但IUT必须至少支持其中一个选项。

1. TRANSMIT SubscribeCOVProperty-Request,

'Subscriber Process Identifier' = (any valid process identifier),

'Monitored Object Identifier' = (any object supporting COV notifications),

'Issue Confirmed Notifications' = TRUE | FALSE,

'Lifetime' = (a value between 60 seconds and 300 seconds),

'Monitored Property Identifier' = (any valid property supporting COV notifications)

1. RECEIVE BACnet-SimpleACK-PDU
2. BEFORE **Notification Fail Time**

IF (the subscription was for confirmed notifications) THEN

RECEIVE BACnetConfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = (the requested subscription lifetime),

'List of Values' = (values appropriate to the property subscribed to,

and any other properties the IUT provides with it,

such as Status-Flags)

TRANSMIT BACnet-SimpleACK-PDU

ELSE

RECEIVE BACnetUnconfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = (the requested subscription lifetime),

'List of Values' = (values appropriate to the property subscribed to, and any other properties the IUT provides with it, such as Status-Flags)

1. MAKE (a change to the monitored object that should cause a COV notification)
2. BEFORE **Notification Fail Time**

IF (the subscription was for confirmed notifications) THEN

RECEIVE BACnetConfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = (a value greater than 0 and less than the requested subscription lifetime),

'List of Values' = (values appropriate to the property subscribed to,

and any other properties the IUT provides with it,

such as Status-Flags)

TRANSMIT BACnet-SimpleACK-PDU

ELSE

RECEIVE BACnetUnconfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = (a value greater than 0 and less than the requested

subscription lifetime),

'List of Values' = (values appropriate to the object type of the monitored object including the changed value that triggered the notification)

1. WAIT (the lifetime of the subscription)
2. MAKE (a change to the monitored object that would cause a COV notification if there were an active subscription)
3. 更新现有订阅

目的：验证IUT是否正确响应SubscribeCOVProperty请求以更新订阅的生命周期。可以使用已确认或未确认通知，但IUT必须至少支持其中一个选项。

测试思想：COV通知的订阅持续60秒。在该订阅到期之前，第二次订阅将持续300秒。 当响应第二个订阅发送通知时，将检查生命周期以验证它是否大于60但小于300秒。

测试步骤：

1. TRANSMIT SubscribeCOVProperty-Request,

'Subscriber Process Identifier' = (any valid process identifier),

'Monitored Object Identifier' = (any object supporting COV notifications),

'Issue Confirmed Notifications' = TRUE | FALSE,

'Lifetime' = 60,

'Monitored Property Identifier' = (any valid property supporting COV notifications)

1. RECEIVE BACnet-SimpleACK-PDU
2. BEFORE **Notification Fail Time**

IF (the subscription was for confirmed notifications) THEN

RECEIVE BACnetConfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = 60,

'List of Values' = (values appropriate to the object type of the

monitored object including the changed value of

that triggered the notification)

TRANSMIT BACnet-SimpleACK-PDU

ELSE

RECEIVE BACnetUnconfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = 60,

'List of Values' = (values appropriate to the object type of the

monitored object including the changed value of

that triggered the notification)

1. TRANSMIT SubscribeCOVProperty-Request,

'Subscriber Process Identifier' = (any valid process identifier),

'Monitored Object Identifier' = (any object supporting COV notifications),

'Issue Confirmed Notifications' = TRUE | FALSE,

'Lifetime' = (a value between 180 and 300 seconds),

'Monitored Property Identifier' = (any valid property supporting COV notifications)

1. RECEIVE BACnet-SimpleACK-PDU
2. BEFORE **Notification Fail Time**

IF (the subscription was for confirmed notifications) THEN

RECEIVE BACnetConfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = (the requested subscription lifetime),

'List of Values' = (values appropriate to the object type of the

monitored object including the changed value of

that triggered the notification)

ELSE

RECEIVE BACnetUnconfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = (the requested subscription lifetime),

'List of Values' = (values appropriate to the object type of the

monitored object including the changed value of

that triggered the notification)

1. 客户端COV增量

目的：当客户端在SubscribeCOVProperty请求中提供COV增量时，验证IUT是否正确生成COV通知。可以使用已确认或未确认的通知，但IUT必须至少支持其中一个选项。

测试思想：对数据类型为REAL的属性进行COV通知订阅。订阅请求指定COV增量。受监视的属性更改小于增量的值，并且TD等待确保IUT不生成通知。被监视的属性的变化量略大于引起COV通知所需的数量，并且TD等待通知。

测试配置：如果订阅的属性在对象中具有相关的COV\_Increment属性，则COV\_Increment属性的值应与订阅服务中提供的COV增量显著不同。

1. TRANSMIT SubscribeCOVProperty-Request,

'Subscriber Process Identifier' = (any valid process identifier),

'Monitored Object Identifier' = (any object supporting COV notifications),

'Issue Confirmed Notifications' = TRUE | FALSE,

'Lifetime' = (any value that will ensure no re-subscription is required to complete the test),

'Monitored Property Identifier' = (any valid property supporting COV notifications),

'COV Increment' = (any valid increment value)

1. RECEIVE BACnet-SimpleACK-PDU
2. BEFORE **Notification Fail Time**

IF (the subscription was for confirmed notifications) THEN

RECEIVE BACnetConfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = (the requested lifetime),

'List of Values' = (values appropriate to the object type of the monitored

object including the value of monitored property)

TRANSMIT BACnet-SimpleACK-PDU

ELSE

RECEIVE BACnetUnconfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = (the requested lifetime),

'List of Values' = (values appropriate to the object type of the monitored

object including the value of monitored property)

1. MAKE (the monitored property change by less than the COV increment)
2. CHECK (verify that the IUT did not transmit a notification message for the monitored property)
3. MAKE (the monitored property change by slightly more than COV Increment less the amount

changed in step 5)

1. BEFORE **Notification Fail Time**

IF (the subscription was for confirmed notifications) THEN

RECEIVE BACnetConfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = ?,

'List of Values' = (values appropriate to the object type of the

monitored object including the changed value that

triggered the notification)

TRANSMIT BACnet-SimpleACK-PDU

ELSE

RECEIVE BACnetUnconfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same identifier used in the subscription),

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = ?,

'List of Values' = (values appropriate to the object type of the

monitored object including the changed value that

triggered the notification)

1. 负响应SubscribeCOVProperty服务执行测试

此测试组的目的是在预期服务失败的情况下验证SubscribeCOVProperty服务请求的正确执行。

1. 受监视对象不支持COV通知

目的：验证当受监视对象不支持COV通知时，IUT正确响应SubscribeCOVProperty请求以建立订阅。

测试步骤：

1. TRANSMIT SubscribeCOVProperty-Request,

'Subscriber Process Identifier' = (any valid process identifier),

'Monitored Object Identifier' = (any object that does not support COV notifications),

'Issue Confirmed Notifications' = TRUE,

'Lifetime' = 60,

'Monitored Property Identifier' = (any property in the object)

1. RECEIVE BACnet-Error-PDU,

Error Class = SERVICES,

Error Code = SERVICE\_REQUEST\_DENIED | OTHER

1. 受监视属性不支持COV通知

目的：验证IUT是否正确响应SubscribeCOVProperty请求，以便在受监视对象支持COV通知但不支持请求的属性时建立订阅。

测试步骤：

1. TRANSMIT SubscribeCOVProperty-Request,

'Subscriber Process Identifier' = (any valid process identifier),

'Monitored Object Identifier' = (any object that supports COV notifications),

'Issue Confirmed Notifications' = TRUE,

'Lifetime' = 60,

'Monitored Property Identifier' = (any property that does not support COV notifications)

1. RECEIVE BACnet-Error PDU,

Error Class = SERVICES,

Error Code = SERVICE\_REQUEST\_DENIED | OTHER

* 1. 原子读文件（AtomicReadFile）服务执行测试

本节定义了证明支持执行AtomicReadFile服务请求所必需的测试。

所需测试：无。

BACnet参考条款：14.1。

测试思想：BACnet文件访问服务允许以记录为基础或以流为基础访问文件。如果IUT支持基于记录的文件结构，则应使用记录访问测试（10.12.1.1和10.12.2.1）。如果IUT支持基于流的文件结构，则应使用流访问测试（10.12.1.2和10.12.2.2）。如果IUT中仅支持一种文件访问类型，则应在PICS中明确说明，并且只需执行本节中适用于该文件类型的测试。测试包括以各种方式使用AtomicReadFile服务读取文件的内容，并验证是否返回了适当的已知文件数据。

配置要求：AtomicReadFile服务执行测试要求TD知道已知文件的确切内容。IUT应配置一个支持记录访问的文件和一个支持流访问的文件。在测试过程中，“R”将指定记录访问测试文件的File对象标识符，“S”将指定流访问测试文件的File对象标识符。如果IUT不支持记录访问和流访问，则可以省略这些文件中的一个。最小测试文件大小为4个八位字节用于流访问，4个记录用于记录访问文件。可以将这些文件配置到IUT中，也可以使用AtomicWriteFile服务将文件初始化为已知状态。测试程序假定IUT已配置有供应商提供的已知文件数据。

1. 正响应AtomicReadFile服务执行测试
2. 读取基于记录的文件

此测试组的目的是在预期服务成功完成的情况下验证从基于记录的文件中正确执行AtomicReadFile服务请求。

1. 读取整个文件

目的：验证IUT是否正确响应读取整个文件的请求。

测试步骤：

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = R,

'File Start Record' = 0,

'Requested Record Count' = (the number of records in the test file)

1. RECEIVE AtomicReadFile-ACK,

'End of File' = TRUE,

'File Start Record' = 0,

'Returned Record Count' = (the number of records in the test file),

'File Record Data' = (the known contents of the test file)

1. 从文件开头读取数据

目的：验证IUT是否正确响应读取从文件开头到文件结束前的中间点数据的请求。

测试步骤：

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = R,

'File Start Record' = 0,

'Requested Record Count' = (any number n: 0 < n < the number of records in the test file)

1. RECEIVE AtomicReadFile-ACK,

'End of File' = FALSE,

'File Start Record' = 0,

'Returned Record Count' = n,

'File Record Data' = (the first n records of the test file)

1. 读取中间点到文件末尾数据

目的：验证IUT是否正确响应从中间点开始并继续到文件末尾的数据读取请求。

测试步骤：

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = R,

'File Start Record' = (any number n: 0 < n < the number of records in the test file),

'Requested Record Count' = (the number of records in the test file – n)

1. RECEIVE AtomicReadFile-ACK,

'End of File' = TRUE,

'File Start Record' = n,

'Returned Record Count' = (the number of records in the test file – n),

'File Record Data' = (the test file record data from position n to the end of the file)

1. 读取从中间点开始并在文件中的另一个中间点结束数据

目的：验证IUT是否正确响应从中间点开始读取数据并到文件中另一个中间点结束的请求。

测试步骤：

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = R,

'File Start Record' = (any number n: 0 < n < (the number of records in the test file – 2)),

'Requested Record Count' = (any number m: 0 < m < the number of records remaining in the test file)

1. RECEIVE AtomicReadFile-ACK,

'End of File' = FALSE,

'File Start Record' = n,

'Returned Record Count' = m,

'File Record Data' = (the specified test file record data)

1. 读取零大小的数据块

目的：验证IUT是否正确响应读取零文件数据记录的请求。

测试步骤：

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = R,

'File Start Record' = (any number n: 0 ≤ n < the number of records in the test file),

'Requested Record Count' = 0

1. RECEIVE AtomicReadFile-ACK,

'End of File' = FALSE,

'File Start Record' = n,

'Returned Record Count' = 0,

'File Record Data' = (an empty list of records)

1. 读取文件末尾的数据

目的：验证IUT是否正确响应从任何点开始读取数据在文件末尾结束的请求。

测试步骤：

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = R,

'File Start Record' = (any number n: 0 ≤ n < the number of records in the test file),

'Requested Record Count' = (any number m: m > the number of records remaining in the test file)

1. RECEIVE AtomicReadFile-ACK,

'End of File' = TRUE,

'File Start Record' = n,

'Returned Record Count' = (the number of records in the test file – n),

'File Record Data' = (the test file records from position n to the end of the file)

1. 读取基于流的文件

此测试组的目的是在预期服务成功完成的情况下验证从基于流的文件正确执行AtomicReadFile服务请求。

1. 读取整个文件

目的：验证IUT是否正确响应读取整个文件的请求。

测试步骤：

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = S,

'File Start Position' = 0,

'Requested Octet Count' = (the number of octets in the test file)

1. RECEIVE AtomicReadFile-ACK,

'End of File' = TRUE,

'File Start Position' = 0,

'File Data' = (the known contents of the test file)

1. 从文件开头读取数据

目的：验证IUT是否正确响应从文件开头开始读取数据到文件结束前的中间点结束的请求。

测试步骤：

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = S,

'File Start Position' = 0,

'Requested Octet Count' = (any number n: 0 < n < the number of octets in the test file)

1. RECEIVE AtomicReadFile-ACK,

'End of File' = FALSE,

'File Start Position' = 0,

'File Data' = (the first n octets of the test file)

1. 读取从中间点到文件末尾的数据

目的：验证IUT是否正确响应从中间点开始并继续到文件末尾的数据读取请求。

测试步骤：

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = S,

'File Start Position' = (any number n: 0 < n < the number of octets in the test file),

'Requested Octet Count' = (the number of octets in the test file – n)

1. RECEIVE AtomicReadFile-ACK,

'End of File' = TRUE,

'File Start Position' = n,

'File Data' = (the test file data from position n to the end of the file)

1. 中间点开始读取数据并在文件中的另一个中间点结束

目的：验证IUT是否正确响应从中间点开始读取数据并继续到文件中另一个中间点的请求。

测试步骤：

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = S,

'File Start Position' = (any number n: 0 < n < (the number of octets in the test file – 2)),

'Requested Octet Count' = (any number m: 0 < m < the number of octets remaining in the test file)

1. RECEIVE AtomicReadFile-ACK,

'End of File' = FALSE,

'File Start Position' = n,

'File Data' = (the specified test file data)

1. 读取零大小的数据块

目的：验证IUT是否正确响应读取零八位字节文件数据的请求。

测试步骤：

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = S,

'File Start Position' = (any number n: 0 ≤ n < the number of octets in the test file),

'Requested Octet Count' = 0

1. RECEIVE AtomicReadFile-ACK,

'End of File' = FALSE,

'File Start Position = n,

'File Data' = (an octet string of length 0)

1. 读取文件末尾的数据

目的：验证IUT是否正确响应从任何点开始读取数据并在文件末尾结束的请求。

测试步骤：

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = S,

'File Start Position' = (any number n: 0 ≤ n < the number of octets in the test file),

'Requested Octet Count' = (any number m: m > the number of octets remaining in the test file)

1. RECEIVE AtomicReadFile-ACK,

'End of File' = TRUE,

'File Start Position' = n,

'File Data' = (the test file octets from position n to the end of the file)

1. 负响应AtomicReadFile服务执行测试
2. 读取基于记录的文件

此测试组的目的是在预期服务失败的情况下验证从基于记录的文件中正确执行AtomicReadFile服务请求。

1. 从文件边界外的一系列记录中读取数据

目的：在指定记录超出文件边界的情况下验证AtomicReadFile服务请求的正确执行。

测试步骤：

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = R,

'File Start Record' = (any number n: n the number of records in the test file),

'Requested Record Count' = (any number > 0)

1. RECEIVE BACnet-Error PDU,

Error Class = SERVICES,

Error Code = INVALID\_FILE\_START\_POSITION

1. 从不存在的文件中读取数据

目的：在指定文件不存在的情况下验证AtomicReadFile服务请求的正确执行。

测试步骤：

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = (any non existent file),

'File Start Record' = (any number 0),

'Requested Record Count' = (any number > 0)

1. RECEIVE BACnet-Error PDU,

Error Class = OBJECT,

Error Code = UNKNOWN\_OBJECT

1. 使用错误的文件访问类型读取数据

目的：在文件访问类型不适合指定文件的情况下，验证AtomicReadFile服务请求的正确执行。

测试步骤：

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = R,

'File Start Position' = 0,

'Requested Octet Count' = 1

1. RECEIVE BACnet-Error PDU,

Error Class = SERVICES,

Error Code = INVALID\_FILE\_ACCESS\_METHOD

1. 从小于零的记录号开始读取数据

目的：在指定记录范围无效的情况下验证AtomicReadFile服务请求的正确执行。

测试步骤：

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = R,

'File Start Record' = (any number n: n <0),

'Requested Record Count' = 1

1. RECEIVE

(BACnet-Error PDU,

Error Class = SERVICES,

Error Code = INVALID\_START\_POSITION) |

(BACnet-Reject-PDU,

Reject Reason = PARAMETER\_OUT\_OF\_RANGE)

1. 读取基于流的文件

此测试组的目的是在预期服务失败的情况下验证从基于流的文件正确执行AtomicReadFile服务请求。

1. 从文件边界外的一系列记录中读取数据

目的：在指定八位字节超出文件边界的情况下验证AtomicReadFile服务请求的正确执行。

测试步骤：

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = S,

'File Start Position' = (any number n: n ≥ the number of octets in the test file),

'Requested Octet Count' = (any number > 0)

1. RECEIVE BACnet-Error PDU,

Error Class = SERVICES,

Error Code = INVALID\_FILE\_START\_POSITION

1. 从不存在的文件中读取数据

目的：在指定文件不存在的情况下验证AtomicReadFile服务请求的正确执行。

测试步骤：

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = (any non existent file),

'File Start Position' = (any number ≥ 0),

'Requested Octet Count' = (any number > 0)

1. RECEIVE BACnet-Error PDU,

Error Class = OBJECT,

Error Code = UNKNOWN\_OBJECT

1. 使用错误的文件访问类型读取数据

目的：在文件访问类型不适合指定文件的情况下，验证AtomicReadFile服务请求的正确执行。

测试步骤：

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = S,

'File Start Record' = 0,

'Requested Record Count' = 1

1. RECEIVE BACnet-Error PDU,

Error Class = SERVICES,

Error Code = INVALID\_FILE\_ACCESS\_METHOD

1. 从小于零的起始位置开始读取数据

目的：在指定记录范围无效的情况下验证AtomicReadFile服务请求的正确执行。

测试步骤：

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = S,

'File Start Position' = (any number n: n < 0),

'Requested Octet Count' = 1

1. RECEIVE

(BACnet-Error PDU,

Error Class = SERVICES,

Error Code = INVALID\_START\_POSITION) |

(BACnet-Reject-PDU,

Reject Reason = PARAMETER\_OUT\_OF\_RANGE)

* 1. 原子写文件（AtomicWriteFile）服务执行测试

本节定义了证明支持执行AtomicWriteFile服务请求所必需的测试。

所需测试：AtomicReadFile服务执行测试，10.12; ReadProperty服务执行测试，10.18。

BACnet参考条款：14.2。

测试思想：BACnet文件访问服务允许以记录为基础或以流为基础访问文件。如果IUT支持对基于记录的文件的写访问，则应使用记录访问测试（10.13.1.1和10.13.2.1）。如果IUT支持基于流的文件结构，则应使用流访问测试（10.13.1.2和10.13.2.2）。如果IUT仅支持其中一种文件访问类型，则应在PICS中明确说明，并且只需执行本节中适用于该文件访问类型的测试。测试内容有以各种方式使用AtomicWriteFile服务修改文件的内容，并验证是否对文件数据进行了正确更改。

其中一些实现可能对文件有特殊限制。例如，表示设备操作软件的文件可能包含用于确保文件真实性的专有标头信息。任何此类特殊限制必须记录在PICS中。

配置要求：如果支持对基于记录的文件的写访问，则IUT应配置一个允许写访问的基于记录的文件对象。该文件的对象标识符将在测试描述中指定为“R”。 如果支持对基于流的文件的写访问，则IUT应配置有允许写访问的基于流的文件对象。该文件的对象标识符将在测试描述中指定为“S”。供应商应提供足够的测试数据或信息写入这些文件，以允许测试人员构建测试数据。文件对象应配置与测试数据不同的初始数据。

1. 正响应AtomicWriteFile服务执行测试
2. 写入基于记录的文件

此测试组的目的是在预期服务成功完成的情况下验证对基于记录的文件的AtomicWriteFile服务请求的正确执行。

1. 编写整个文件

目的：验证IUT是否正确响应写入整个文件的请求。

配置要求：测试数据应包含至少与文件初始数据一样多的记录。

测试步骤：

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = R,

'File Start Record' = 0,

'Requested Record Count' = (any number ≥ the number of records in the test data)

1. RECEIVE AtomicReadFile-ACK,

'End of File' = TRUE,

'File Start Record' = 0,

'Returned Record Count' = (any number ≤ the number of records in the test data),

'File Record Data' = (the initial data)

1. TRANSMIT AtomicWriteFile-Request,

'File Identifier' = R,

'File Start Record' = 0,

'Record Count' = (the number of records in the test data),

'File Record Data' = (the test data)

1. RECEIVE AtomicWriteFile-ACK,

'File Start Record' = 0

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = R,

'File Start Record' = 0,

'Requested Record Count' = (any number > the number of records in the test data)

1. RECEIVE AtomicReadFile-ACK,

'End of File' = TRUE,

'File Start Record' = 0,

'Returned Record Count' = (the number of records in the test data),

'File Record Data' = (the test data)

1. VERIFY (R), Modification\_Date = (the current date and time)
2. VERIFY (R), ARCHIVE = FALSE
3. VERIFY (R), Number\_Of\_Records = (the number of records in the test data)
4. 覆盖部分文件

目的：验证IUT是否正确响应从任何中间点开始写入文件的请求。 如果IUT不支持除了替换整个文件之外无法修改的文件，并且在PICS中明确说明了此限制，则可以忽略此测试。

配置要求：文件对象应配置与测试数据不同的数据。

测试步骤：

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = R,

'Property Identifier' = Number\_Of\_Records

1. RECEIVE ReadProperty-ACK,

'Object Identifier' = R,

'Property Identifier' = Number\_Of\_Records

'Property Value' = (the current number of records, designated "InitialNumRecords" below)

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = R,

'File Start Record' = 0,

'Requested Record Count' = (any number > InitialNumRecords)

1. RECEIVE AtomicReadFile-ACK,

'End of File' = TRUE,

'File Start Record' = 0,

'Returned Record Count' = InitialNumRecords,

'File Record Data' = (the initial data)

1. TRANSMIT AtomicWriteFile-Request,

'File Identifier' = R,

'File Start Record' = (any value n: 0 < n < InitialNumRecords),

'Record Count' = (the number of records in the test data),

'File Record Data' = (the test data)

1. RECEIVE AtomicWriteFile-ACK,

'File Start Record' = (the 'File Start Record' used in step 4)

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = R,

'File Start Record' = (the 'File Start Record' used in step 4),

'Requested Record Count' = (the number of records in the test data)

1. RECEIVE AtomicReadFile-ACK,

'End of File' = TRUE,

'File Start Record' = (the 'File Start Record' used in step 4),

'Returned Record Count' = (the number of records in the test data),

'File Record Data' = (the test data)

1. VERIFY (R), Modification\_Date = (the current date and time)
2. VERIFY (R), ARCHIVE = FALSE
3. VERIFY (R), Number\_Of\_Records = (the number of records in the test data + the 'FileStart Record'

used in step 4)

1. 将数据附加到文件末尾

目的：验证IUT是否正确响应写入文件末尾的请求。如果IUT不支持除了替换整个文件之外无法修改的文件，并且在PICS中明确说明了此限制，则忽略此测试。

配置要求：文件对象应配置与测试数据不同的数据。

测试步骤：

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = R,

'Property Identifier' = Number\_Of\_Records

1. RECEIVE ReadProperty-ACK,

'Object Identifier' = R,

'Property Identifier' = Number\_Of\_Records

'Property Value' = (the current number of records, designated "InitialNumRecords" below)

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = R,

'File Start Record' = 0,

'Requested Record Count' = (any number > InitialNumRecords)

1. RECEIVE AtomicReadFile-ACK,

'End of File' = TRUE,

'File Start Record' = 0,

'Returned Record Count' = InitialNumRecords,

'Returned Record Count' = InitialNumRecords,

1. TRANSMIT AtomicWriteFile-Request,

'File Identifier' = R,

'File Start Record' = -1,

'Record Count' = (the number of records in the test data),

'File Record Data' = (the test data)

1. RECEIVE AtomicWriteFile-ACK,

'File Start Record' = InitialNumRecords,

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = R,

'File Start Record' = InitialNumRecords,

'Requested Record Count' = (the number of records in the test data)

1. RECEIVE AtomicReadFile-ACK,

'End of File' = TRUE,

'File Start Record' = InitialNumRecords,

'Returned Record Count' = (the number of records in the test data),

'File Record Data' = (the test data)

1. VERIFY (R), Modification\_Date = (the current date and time)
2. VERIFY (R), ARCHIVE = FALSE
3. VERIFY (R), Number\_Of\_Records = (the number of records in the test data +InitialNumRecords)
4. 截断文件

目的：验证IUT是否正确响应WriteProperty请求以截断文件。仅当Device对象中存在Protocol\_Revision属性且值大于或等于1时，才应执行此测试。

所需测试：WriteProperty服务执行测试，10.22。

配置要求：供应商应使用允许写访问并包含长度超过一个记录的初始文件数据的文件对象来配置IUT。还应向测试人员提供文件数据的副本。

测试步骤：

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = R,

'Property Identifier' = Number\_Of\_Records

1. RECEIVE ReadProperty-ACK,

'Object Identifier' = R,

'Property Identifier' = Number\_Of\_Records

'Property Value' = (the current number of records, designated "InitialNumRecords" below)

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = R,

'File Start Record' = 0,

'Requested Record Count' = (any number > InitialNumRecords)

1. RECEIVE AtomicReadFile-ACK,

'End of File' = TRUE,

'File Start Record' = 0,

'Returned Record Count' = InitialNumRecords,

'File Record Data' = (the initial data)

1. TRANSMIT WriteProperty-Request,

'Object Identifier' = R,

'Property Identifier' = Number\_Of\_Records,

'Property Value' = (any value n: 0 < n < InitialNumRecords)

1. RECEIVE BACnet-SimpleACK-PDU
2. TRANSMIT AtomicReadFile-Request,

'File Identifier' = R,

'File Start Record' = 0,

'Requested Record Count' = InitialNumRecords

1. RECEIVE AtomicReadFile-ACK,

'End of File' = TRUE,

'File Start Record' = 0,

'Returned Record Count' = n,

'File Record Data' = (the initial data from records 0 – (n-1))

1. 删除文件

目的：验证IUT是否正确响应WriteProperty请求以删除文件。仅当Device对象中存在Protocol\_Revision属性且值大于或等于1时，才应执行此测试。

所需测试：WriteProperty服务执行测试，10.22。

测试步骤：

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = R,

'Property Identifier' = Number\_Of\_Records

1. RECEIVE ReadProperty-ACK,

'Object Identifier' = R,

'Property Identifier' = Number\_Of\_Records

'Property Value' = the current number of records, designated "InitialNumRecords" below)

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = R,

'File Start Record' = 0,

'Requested Record Count' = (any number > InitialNumRecords)

1. RECEIVE AtomicReadFile-ACK,

'End of File' = TRUE,

'File Start Record' = 0,

'Returned Record Count' = InitialNumRecords,

'File Record Data' = (the initial data for this file)

1. TRANSMIT WriteProperty-Request,

'Object Identifier' = R,

'Property Identifier' = Number\_Of\_Records,

'Property Value' = 0

1. RECEIVE BACnet-SimpleACK-PDU
2. TRANSMIT AtomicReadFile-Request,

'File Identifier' = R,

'File Start Record' = 0,

'Requested Record Count' = InitialNumRecords

1. RECEIVE AtomicReadFile-ACK,

'End of File' = TRUE,

'File Start Record' = 0,

'Returned Record Count' = 0,

'File Record Data' = (an empty list of records)

1. 写入基于流的文件

此测试组的目的是在预期服务成功完成的情况下验证对基于流的文件的AtomicWriteFile服务请求的正确执行。

1. 编写整个文件

目的：验证IUT是否正确响应写入整个文件的请求。

配置要求：测试数据应至少包含与文件初始数据一样多的八位字节。

测试步骤：

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = S,

'File Start Position = 0,

'Requested Octet Count' = (any number ≥ the number of octets in the test data)

1. RECEIVE AtomicReadFile-ACK,

'End of File' = TRUE,

'File Start Position' = 0,

'File Record Data' = (the initial data)

1. TRANSMIT AtomicWriteFile-Request,

'File Identifier' = S,

'File Start Position = 0,

'File Data' = (the test data)

1. RECEIVE AtomicWriteFile-ACK,

'File Start Position' = 0

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = S,

'File Start Position' = 0,

'Requested Octet Count' = (any number > the number of octets in the test data)

1. RECEIVE AtomicReadFile-ACK,

'End of File' = TRUE,

'File Start Position' = 0,

'File Data' = (the test data)

1. VERIFY (R), Modification\_Date = (the current date and time)
2. VERIFY (R), ARCHIVE = FALSE
3. VERIFY (R), Number\_Of\_Records = (the number of records in the test data)
4. 覆盖部分文件

目的：验证IUT是否正确响应从中间点开始写入文件的请求。如果IUT不支持只能通过替换整个文件进行修改的文件，并且在PICS中明确说明了此限制，则可以忽略此测试。

配置要求：文件对象应配置与测试数据不同的数据。

测试步骤：

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = S,

'Property Identifier' = File\_Size

1. RECEIVE ReadProperty-ACK,

'Object Identifier' = S,

'Property Identifier' = File\_Size,

'Property Value' = (the current file size, designated "InitialNumOctets" below)

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = S,

'File Start Position' = 0,

'Requested Octet Count' = (any number > InitialNumOctets)

1. RECEIVE AtomicReadFile-ACK,

'End of File' = TRUE,

'File Start Position' = 0,

'File Data' = (the initial data)

1. TRANSMIT AtomicWriteFile-Request,

'File Identifier' = S,

'File Start Position' = (any value n: 0 < n < InitialNumOctets),

'File Record Data' = (the test data)

1. RECEIVE AtomicWriteFile-ACK,

'File Start Position = (the 'File Start Position' used in step 4)

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = S,

'File Start Position' = (the 'File Start Position' used in step 4),

'Requested Octet Count' = (the number of octets in the test data)

1. RECEIVE AtomicReadFile-ACK,

'End of File' = TRUE,

'File Start Position' = (the 'File Start Position' used in step 4),

'File Data' = (the test data)

1. VERIFY (R), Modification\_Date = (the current date and time)
2. VERIFY (R), ARCHIVE = FALSE
3. VERIFY (R), File\_Size = (the number of octets in the test data + the 'File Start Position'

used in step 4)

1. 将数据附加到文件末尾

目的：验证IUT是否正确响应写入文件末尾的请求。如果IUT不支持只能通过替换整个文件进行修改的文件，并且在PICS中明确说明了此限制，则可以忽略此测试。

配置要求：文件对象应配置与测试数据不同的初始数据。

测试步骤：

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = S,

'Property Identifier' = File\_Size

1. RECEIVE ReadProperty-ACK,

'Object Identifier' = S,

'Property Identifier' = File\_Size,

'Property Value' = (the current size in octets, designated "InitialNumOctets" below)

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = S,

'File Start Position' = 0,

'Requested Octet Count' = (any number > InitialNumOctets)

1. RECEIVE AtomicReadFile-ACK,

'End of File' = TRUE,

'File Start Position' = 0,

'File Data' = (the initial data)

1. TRANSMIT AtomicWriteFile-Request,

'File Identifier' = S,

'File Start Position = -1,

'File Data' = (the test data)

1. RECEIVE AtomicWriteFile-ACK,

'File Start Position = InitialNumOctets,

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = S,

'File Start Position' = InitialNumOctets,

'Requested Octet Count' = (the number of octets in the test data)

1. RECEIVE AtomicReadFile-ACK,

'End of File' = TRUE,

'File Start Position' = InitialNumOctets,

'File Data' = (the test data)

1. VERIFY (R), Modification\_Date = (the current date and time)
2. VERIFY (R), ARCHIVE = FALSE
3. VERIFY (R), File\_Size = (the number of octets in the test data + InitialNumOctets)
4. 截断文件

目的：验证IUT是否正确响应WriteProperty请求以截断文件。仅当Device对象中存在Protocol\_Revision属性且值大于或等于1时，才应执行此测试。

所需测试：WriteProperty服务执行测试10.22。

配置要求：供应商应使用允许写入访问并包含长度超过一个八位位组的初始文件数据的文件对象来配置IUT。还应向测试人员提供文件数据的副本。

测试步骤：

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = S,

'Property Identifier' = File\_Size

1. RECEIVE ReadProperty-ACK,

'Object Identifier' = S,

'Property Identifier' = File\_Size,

'Property Value' = the current size in octets, designated "InitialNumOctets" below)

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = S,

'File Start Position' = 0,

'Requested Octet Count' = (any number > InitialNumOctets)

1. RECEIVE AtomicReadFile-ACK,

'End of File' = TRUE,

'File Start Position' = 0,

'File Data' = (the initial data)

1. TRANSMIT WriteProperty-Request,

'Object Identifier' = S,

'Property Identifier' = File\_Size,

'Property Value' = ((any value n: 0< n < InitialNumOctets)

1. RECEIVE BACnet-SimpleACK-PDU
2. TRANSMIT AtomicReadFile-Request,

'File Identifier' = S,

'File Start Position' = 0,

'Requested Octet Count' = InitialNumOctets

1. RECEIVE AtomicReadFile-ACK,

'End of File' = TRUE,

'File Start Position' = 0,

'File Data' = (the test data for this file from octets 0 – (n-1))

1. 删除文件

目的：验证IUT是否正确响应WriteProperty请求以删除文件。仅当Device对象中存在Protocol\_Revision属性且值大于或等于1时，才应执行此测试。

所需测试：WriteProperty服务执行测试10.22。

测试步骤：

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = S,

'Property Identifier' = File\_Size

1. RECEIVE ReadProperty-ACK,

'Object Identifier' = S,

'Property Identifier' = File\_Size

'Property Value' = (the current size in octets, designated "InitialNumOctets" below)

1. TRANSMIT AtomicReadFile-Request,

'File Identifier' = S,

'File Start Position' = 0,

'Requested Octet Count' = (any number > InitialNumOctets)

1. RECEIVE AtomicReadFile-ACK,

'End of File' = TRUE,

'File Start Position' = 0,

'File Data' = (the initial data)

1. TRANSMIT WriteProperty-Request,

'Object Identifier' = S,

'Property Identifier' = File\_Size,

'Property Value' = 0

1. RECEIVE BACnet-SimpleACK-PDU
2. TRANSMIT AtomicReadFile-Request,

'File Identifier' = S,

'File Start Position' = 0,

'Requested Octet Count' = InitialNumOctets

1. RECEIVE AtomicReadFile-ACK,

'End of File' = TRUE,

'File Start Position' = 0,

'File Record Data' = (an octet string with length 0)

1. 负响应AtomicWriteFile服务执行测试

此测试组的目的是在预期服务失败的情况下验证AtomicWriteFile服务请求的正确执行。

1. 写入记录访问文件
2. 使用流访问写入记录访问文件

目的：使用错误的访问方法写入文件，验证是否返回了相应的错误。 仅当IUT支持使用记录访问的文件对象时，才应执行此测试用例。

测试步骤：

1. TRANSMIT AtomicWriteFile-Request,

'File Identifier' = R,

'File Start Position' = 0,

'File Data' = (any stream file data)

1. RECEIVE BACnet-Error-PDU,

Error Class = SERVICES,

Error Code = INVALID\_FILE\_ACCESS\_METHOD

1. 无效起始位置写入文件

目的：使用无效的起始位置写入文件，验证是否返回了相应的错误。

测试步骤：

1. TRANSMIT AtomicWriteFile-Request,

'File Identifier' = R,

'File Start Record' = (any value n: n < -1 | n > the maximum supported number of records),

'Record Count' = (any value > 0),

'File Record Data' = (any record data)

1. RECEIVE BACnet-Error-PDU,

Error Class = SERVICES,

Error Code = INVALID\_FILE\_START\_POSITION

1. 写入只读文件

目的：写入只读文件，验证是否返回了相应的错误。

配置要求：IUT应配置一个不允许写访问的文件。如果无法配置此类文件，则省略此测试。

测试步骤：

1. TRANSMIT AtomicWriteFile-Request,

'File Identifier' = (any record access file that is read only),

'File Start Record' = 0,

'Record Count' = (any value > 0),

'File Record Data' = (any record data)

1. RECEIVE

(BACnet-Error-PDU,

Error Class = PROPERTY,

Error Code = WRITE\_ACCESS\_DENIED) |

(BACnet-Error-PDU,

Error Class = SERVICES,

Error Code = FILE\_ACCESS\_DENIED)

1. 写入不存在的文件

目的：写入不存在的文件，验证是否返回了相应的错误。

测试步骤：

1. TRANSMIT AtomicWriteFile-Request,

'File Identifier' = (any nonexistent file),

'File Start Record' = 0,

'Record Count' = (any value > 0),

'File Record Data' = (any record data)

1. RECEIVE BACnet-Error-PDU,

Error Class = OBJECT,

Error Code = UNKNOWN\_OBJECT | NO\_OBJECTS\_OF\_SPECIFIED\_TYPE

1. 写入流访问文件
2. 使用记录访问写入流访问文件

目的：使用错误的访问方法写入文件，则验证是否返回了相应的错误。只在IUT支持使用流访问的文件对象时，才执行此测试。

测试步骤：

1. TRANSMIT AtomicWriteFile-Request,

'File Identifier' = S,

'File Start Record' = 0,

'Record Count' = (the number of records in the test data),

'File Record Data' = (any record file data)

1. RECEIVE BACnet-Error-PDU,

Error Class = SERVICES,

Error Code = INVALID\_FILE\_ACCESS\_METHOD

1. 写入具有无效起始位置的文件

目的：使用无效的起始位置写入文件，则验证是否返回了相应的错误。

测试步骤：

1. TRANSMIT AtomicWriteFile-Request,

'File Identifier' = S,

'File Start Position' = (any value n: n < -1 | n > the maximum supported number of octets),

'File Data' = (any record data)

1. RECEIVE BACnet-Error-PDU,

Error Class = SERVICES,

Error Code = INVALID\_FILE\_START\_POSITION

1. 写入只读文件

目的：写入只读文件，验证是否返回了相应的错误。

配置要求：IUT应配置一个不允许写访问的文件。如果无法配置此类文件，则省略此测试。

测试步骤：

1. TRANSMIT AtomicWriteFile-Request,

'File Identifier' = (any stream access file that is read only),

'File Start Position' = 0,

'File Data' = (any stream data)

1. RECEIVE

(BACnet-Error-PDU,

Error Class = PROPERTY,

Error Code = WRITE\_ACCESS\_DENIED) |

(BACnet-Error-PDU,

Error Class = SERVICES,

Error Code = FILE\_ACCESS\_DENIED)

1. 写入不存在的文件

目的：写入不存在的文件，验证是否返回相应的错误。

测试步骤：

1. TRANSMIT AtomicWriteFile-Request,

'File Identifier' = (any nonexistent file),

'File Start Position' = 0,

'File Data' = (any stream data)

1. RECEIVE BACnet-Error-PDU,

Error Class = OBJECT,

Error Code = UNKNOWN\_OBJECT | NO\_OBJECTS\_OF\_SPECIFIED\_TYPE

* 1. 添加列表元素（AddListElement）服务执行测试

本节定义了证明支持执行AddListElement服务请求所必需的测试。

所需测试：ReadProperty服务执行测试，10.18。

BACnet参考条款15.1。

配置要求：IUT配置至少一个标准对象，该对象包含数据类型为列表的属性。名称“L”表示测试描述中该对象的对象标识符。该列表属性在测试描述中指定为“ListProp”，应包含一个或多个数据元素，并能够存储其他数据元素。可以使用AddListElement服务更改属性值。

1. 正响应AddListElement服务执行测试

此测试组的目的是在预期服务成功完成的情况下验证AddListElement服务请求的正确执行。

1. 添加单个元素

目的：验证IUT响应AddListElement服务请求以将单个元素添加到列表的能力。

测试步骤：

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = L,

'Property Identifier' = ListProp

1. RECEIVE ReadProperty-ACK,

'Object Identifier' = L,

'Property Identifier' = ListProp,

'Property Value' = (any valid value referred to as "InitialList" below)

1. TRANSMIT AddListElement-Request,

'Object Identifier' = L,

'Property Identifier' = ListProp

'List of Elements' = (a single element of the correct datatype that is not in InitialList)

1. RECEIVE BACnet-Simple-ACK-PDU
2. VERIFY (L), ListProp = (a list containing the elements of InitialList + the newly added element)
3. 添加多个元素

目的：验证IUT响应AddListElement服务请求以将多个元素添加到列表的能力。

测试步骤：

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = L,

'Property Identifier' = ListProp

1. RECEIVE ReadProperty-ACK,

'Object Identifier' = L,

'Property Identifier' = ListProp,

'Property Value' = (any valid value referred to as "InitialList" below)

1. TRANSMIT AddListElement-Request,

'Object Identifier' = L,

'Property Identifier' = ListProp

'List of Elements' = (two elements of the correct datatype that are not in InitialList)

1. RECEIVE BACnet-Simple-ACK-PDU
2. VERIFY (L), ListProp = (a list containing the elements of InitialList + the newly added elements)
3. 添加冗余元素

目的：验证IUT响应AddListElement服务请求以将冗余元素添加到列表的能力。

测试步骤：

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = L,

'Property Identifier' = ListProp

1. RECEIVE ReadProperty-ACK,

'Object Identifier' = L,

'Property Identifier' = ListProp,

'Property Value' = (any valid value referred to as "InitialList" below)

1. TRANSMIT AddListElement-Request,

'Object Identifier' = L,

'Property Identifier' = ListProp

'List of Elements' = (a single element that is already contained in InitialList)

1. RECEIVE BACnet-Simple-ACK-PDU
2. VERIFY (L) ListProp = InitialList
3. 负响应AddListElement服务执行测试

此测试组的目的是在预期服务失败的情况下验证AddListElement服务请求的正确执行。

1. 将列表元素添加到非列表的属性

目的：验证IUT正确响应AddListElement服务请求以将元素添加到非列表属性的能力。

测试步骤：

1. TRANSMIT AddListElement-Request,

'Object Identifier' = (any supported object),

'Property Identifier' = (any writable property that is not a list),

'List of Elements' = (a single element of the same datatype as the specified property)

1. RECEIVE AddListElement-Error,

Error Class = SERVICES,

Error Code = PROPERTY\_IS\_NOT\_A\_LIST,

'First Failed Element' = 0

1. 添加具有无效数据类型的列表元素

目的：验证IUT正确响应AddListElement服务请求以将具有无效数据类型的元素添加到列表的能力。

测试步骤：

1. TRANSMIT AddListElement-Request,

'Object Identifier' = L,

'Property Identifier' = ListProp,

'List of Elements' = (a single element with a datatype inappropriate for this property)

1. RECEIVE AddListElement-Error,

Error Class = PROPERTY,

Error Code = INVALID\_DATATYPE,

'First Failed Element' = 0

1. AddListElement在列表中部分失败

目的：验证IUT响应AddListElement服务请求以将多个元素添加到无法添加其中一个元素的列表的能力。若失败，AddListElement服务应保持列表不变。

测试步骤：

1. READ InitialList = (L), ListProp
2. TRANSMIT AddListElement-Request,

'Object Identifier' = L,

'Property Identifier' = ListProp

'List of Elements' = (two or more elements to be added to the list with the second element having the wrong datatype)

1. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 7) THEN

RECEIVE AddListElement-Error,

Error Class = PROPERTY,

Error Code = INVALID\_DATATYPE

'First Failed Element' = 2

ELSE

RECEIVE AddListElement-Error,

Error Class = SERVICES,

Error Code = INVALID\_PARAMETER\_DATATYPE

'First Failed Element' = 2

1. VERIFY (L), ListProp = InitialList
   1. 移除列表元素（RemoveListElement）服务执行测试

本节定义了演示支持执行RemoveListElement服务请求所必需的测试。

所需测试：ReadProperty服务执行测试，10.18。

BACnet参考条款15.2。

配置要求：IUT应配置至少一个标准对象，该对象包含数据类型为列表的属性。 名称“L”表示测试描述中该对象的对象标识符。此列表属性在测试描述中指定为“ListProp”，应包含两个或多个数据元素，可以使用RemoveListElement服务更改属性值。

1. 正响应RemoveListElement服务执行测试

此测试组的目的是在预期服务成功完成的情况下验证RemoveListElement服务请求的正确执行。

1. 从列表中删除单个元素

目的：验证IUT响应RemoveListElement服务请求以从列表中删除单个元素的能力。

测试步骤：

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = L,

'Property Identifier' = ListProp

1. RECEIVE ReadProperty-ACK,

'Object Identifier' = L,

'Property Identifier' = ListProp,

'Property Value' = (any valid value referred to as "InitialList" below)

1. TRANSMIT RemoveListElement-Request,

'Object Identifier' = L,

'Property Identifier' = ListProp

'List of Elements' = (a single element of InitialList)

1. RECEIVE BACnet-Simple-ACK-PDU
2. VERIFY (L), ListProp = (a list containing all of the elements of InitialList except the one removed in

step 3)

1. 从列表中删除多个元素

目的：验证IUT响应RemoveListElement服务请求以从列表中删除多个元素的能力。

测试步骤：

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = L,

'Property Identifier' = ListProp

1. RECEIVE ReadProperty-ACK,

'Object Identifier' = L,

'Property Identifier' = ListProp,

'Property Value' = (any valid value referred to as "InitialList" below)

1. TRANSMIT RemoveListElement-Request,

'Object Identifier' = L,

'Property Identifier' = ListProp

'List of Elements' = (two or more elements of InitialList)

1. RECEIVE BACnet-Simple-ACK-PDU
2. VERIFY (L), ListProp = (a list containing all of the elements of InitialList except the ones removed in step3)
3. 负响应RemoveListElement服务执行测试

此测试组的目的是在预期服务失败的情况下验证RemoveListElement服务请求的正确执行。

1. 从非列表的属性中删除列表元素

目的：验证IUT正确响应RemoveListElement服务请求以从非列表属性中删除元素的能力。

测试步骤：

1. TRANSMIT RemoveListElement-Request,

'Object Identifier' = (any supported object),

'Property Identifier' = (any writable property that is not a list),

'List of Elements' = (a single element of the same datatype as the specified property)

1. RECEIVE RemoveListElement-Error,

Error Class = SERVICES,

Error Code = PROPERTY\_IS\_NOT\_A\_LIST,

'First Failed Element' = 0

1. RemoveListElement在列表中部分失败

目的：验证IUT响应RemoveListElement服务请求，以从无法删除其中一个元素的列表中删除多个元素的能力。失败时，RemoveListElement服务应使列表保持不变。

测试步骤：

1. READ InitialList = (L), ListProp
2. TRANSMIT RemoveListElement-Request,

'Object Identifier' = L,

'Property Identifier' = ListProp

'List of Elements' = (one element from InitialList, followed by an element of the correct datatype that is not in InitialList, followed by one or more Elements from InitialList)

1. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 7) THEN

RemoveListElement-Error,

Error Class = PROPERTY,

Error Code = INVALID\_DATA\_TYPE

'First Failed Element' =2

ELSE

RECEIVE RemoveListElement-Error,

Error Class = SERVICES | PROPERTY,

Error Code = OTHER

'First Failed Element' =2

1. VERIFY (L), ListProp = InitialList
   1. 创建对象（CreateObject）服务执行测试

所需测试：ReadProperty服务执行测试，10.18。

BACnet参考条款：15.3。

在本节中定义的每个测试中，都有一个步骤，读取Device对象的Object\_List列表属性，以验证新创建的对象是否出现在列表中。如果Object\_List长，则此过程可能不适用于不支持分段的实现。 在这种情况下，可用一次读取Object\_List数组的每个元素，直到找到包含新创建对象的条目为止的方法代替。

1. 正响应CreateObject服务执行测试

测试思想：此部分定义了证明支持执行CreateObject服务请求所必需的测试。BACnet未指定哪些对象类型可动态创建。供应商可以自由地创建他们想要的对象类型的任何组合。这里描述的测试过程是通用的，它们可以应用于任何对象类型，包括专有对象类型。 IUT必须通过将所有这些测试应用于每种类型的一个对象来证明CreateObject服务对于PICS声称可动态创建的每个对象类型都能正常工作。

1. 指定没有初始值的对象类型来创建对象

目的：当使用对象类型作为对象说明符时，验证CreateObject服务请求的正确执行。

测试步骤：

1. READ X1 = Object\_List
2. TRANSMIT CreateObject-Request,

'Object Specifier' = (any creatable object type)

1. RECEIVE CreateObject-ACK,

'Object Identifier' = (O1, any object identifier of the specified type)

1. CHECK (X1 does not contain O1)
2. READ X2 = Object\_List
3. CHECK (X2 contains O1)
4. VERIFY (O1), (any required property of the specified object) =

(any value of the correct datatype for the specified property)

1. 指定没有初始值的对象标识符来创建对象

目的：当对象标识符用作对象说明符时，验证CreateObject服务请求的正确执行。

测试步骤：

1. TRANSMIT CreateObject-Request,

'Object Specifier' = (any unique object identifier of a type that is creatable )

1. RECEIVE CreateObject-ACK,

'Object Identifier' = (the object identifier specified in step 1)

1. VERIFY (the object identifier of the newly created object),

(any required property of the specified object) = (any value of the correct datatype for the specified property)

1. VERIFY (the IUT's Device object), Object\_List = (any object list containing the newly created object)
2. 通过指定对象类型和提供初始值来创建对象

目的：当使用对象类型作为对象说明符并提供初始属性值列表时，验证CreateObject服务请求的正确执行。

测试步骤：

1. READ X1 = Object\_List
2. TRANSMIT CreateObject-Request,

'Object Specifier' = (any creatable object type),

'List Of Initial Values' = (a list of one or more properties and their initial values that the IUT will accept)

1. RECEIVE CreateObject-ACK,

'Object Identifier' = (any unique object identifier of the specified type)

1. CHECK (X1 does not contain O1)
2. READ X2 = Object\_List
3. CHECK (X2 contains O1)
4. REPEAT X = (properties initialized in the CreateObject-Request) DO {

VERIFY (O1), X = (the value specified in the 'List Of Initial Values' parameter of the

CreateObject-Request)

}

1. 通过指定对象标识符并提供初始值来创建对象

目的：当对象标识符用作对象说明符并提供初始属性值列表时，验证CreateObject服务请求的正确执行。

测试步骤：

1. TRANSMIT CreateObject-Request,

'Object Specifier' = (any unique object identifier of a type that is creatable )

'List Of Initial Values' = (a list of one or more properties and their initial values that the IUT will accept)

1. RECEIVE CreateObject-ACK,

'Object Identifier' = (the object identifier specified in step 1)

1. REPEAT X = (properties initialized in the CreateObject-Request) DO {

VERIFY (the object identifier for the newly created object),

X = (the value specified in the 'List Of Initial Values' parameter of the CreateObject-Request)

}

1. VERIFY (the IUT's Device object), Object\_List = (any object list containing the newly created object)
2. 负响应CreateObject服务执行测试

此测试组的目的是在预期服务失败的情况下验证CreateObject服务请求的正确执行。

1. 尝试创建没有唯一对象标识符的对象

目的：当“Object Specifier”参数传送IUT中已存在的对象标识符时，验证CreateObject服务请求的正确执行。

测试步骤：

1. TRANSMIT CreateObject-Request,

'Object Identifier' = (any object identifier representing an object that already exists having an object type for which dynamic creation is supported)

1. RECEIVE CreateObject-Error,

Error Class = OBJECT,

Error Code = OBJECT\_IDENTIFIER\_ALREADY\_EXISTS

'First Failed Element Number' = 0

1. 尝试通过指定对象类型创建具有不可创建的对象类型的对象

目的：当“对象说明符”参数传送在IUT中不可动态创建的对象类型时，验证CreateObject服务请求的正确执行。

测试步骤

1. TRANSMIT CreateObject-Request,

'Object Type' = (any supported object type for which dynamic creation is not supported)

1. RECEIVE CreateObject-Error,

Error Class = OBJECT,

Error Code = DYNAMIC\_CREATION\_NOT\_SUPPORTED

'First Failed Element Number' = 0

1. 尝试通过指定对象标识符来创建具有不可创建的对象标识符的对象

目的：当“Object Specifier”参数传送IUT中不可动态创建的对象类型的对象标识符时，验证CreateObject服务请求的正确执行。

测试步骤：

1. TRANSMIT CreateObject-Request,

'Object Identifier' = (any object identifier having a supported object type for which dynamic creation is not supported)

1. RECEIVE CreateObject-Error,

Error Class = OBJECT,

Error Code = DYNAMIC\_CREATION\_NOT\_SUPPORTED

'First Failed Element Number' = 0

1. VERIFY(the IUT’s Device object),

Object\_List = (any object list that does not contain the object specified in step 1)

注意事项：如果IUT限制了可以创建的实例，在步骤1中选择对象标识符时应考虑到这一点。

1. 尝试使用对象类型说明符和初始值中的错误创建对象

目的：当对象类型用作对象说明符并且提供包含无效值的初始属性值列表时，验证CreateObject服务请求的正确执行。

测试步骤：

1. READ X1 = Object\_List
2. TRANSMIT CreateObject-Request,

'Object Type' = (any creatable object type),

'List Of Initial Values' = (a list of one or more propertiesand their initial values, that the IUT will accept initial values for, with one of the values being out of range)

1. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 4) THEN

RECEIVE CreateObject-Error PDU,

Error Class = PROPERTY,

Error Code = VALUE\_OUT\_OF\_RANGE

'First Failed Element Number' = (the position in the 'List Of Initial Values' with the offending value)

ELSE

RECEIVE CreateObject-Error,

Error Class = PROPERTY,

Error Code = VALUE\_OUT\_OF\_RANGE | OTHER

'First Failed Element Number' = (the position in the 'List Of Initial Values' with the offending value)

1. CHECK（Verify that the new object was not created）
2. TRANSMIT CreateObject-Request,

'Object Type' = (object type from step 2),

'List Of Initial Values' = (a list of one or more properties and their initial values, that the IUT will accept initial values for,with one of the values being an inappropriate datatype)

1. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 4) THEN

RECEIVE CreateObject-Error,

Error Class = PROPERTY,

Error Code = INVALID\_DATATYPE

'First Failed Element Number' = (the position in the 'List Of Initial Values' with the offending value)

ELSE

RECEIVE CreateObject-Error,

Error Class = PROPERTY,

Error Code = VALUE\_OUT\_OF\_RANGE | INVALID\_DATATYPE | OTHER

'First Failed Element Number' = (the position in the 'List Of Initial Values' with the offending value)

| (BACnet-Reject-PDU

Reject Reason = INVALID\_PARAMETER\_DATATYPE |

INVALID\_TAG)

1. READ X2 = Object\_List
2. CHECK(X1 = X2)
3. 尝试使用对象标识符对象说明符和初始值中的错误创建对象

目的：当对象标识符用作对象说明符并且提供包含无效值的初始属性值列表时，验证CreateObject服务请求的正确执行。

测试步骤：

1. TRANSMIT CreateObject-Request,

'Object Identifier' = (any unique object identifier of a type that is creatable and an instance number that is creatable),

'List Of Initial Values' = (a list of one or more properties and their initial values,that the IUT will accept initial values for, with one of the values being out of range)

1. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 4) THEN

RECEIVE CreateObject-Error PDU,

Error Class = PROPERTY,

Error Code = VALUE\_OUT\_OF\_RANGE

'First Failed Element Number' = (the position in the 'List Of Initial Values' with the offending value)

ELSE

RECEIVE CreateObject-Error,

Error Class = PROPERTY,

Error Code = VALUE\_OUT\_OF\_RANGE | OTHER

'First Failed Element Number' = (the position in the 'List Of Initial Values' with the offending value)

1. CHECK (Verify that the new object was not created)
2. TRANSMIT CreateObject-Request,

'Object Identifier' = (object identifier from step 1),

'List Of Initial Values' = (a list of one or more properties and their initial values, that the IUT will accept initial values for,with one of the values being an inappropriate datatype)

1. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 4) THEN

RECEIVE CreateObject-Error,

Error Class = PROPERTY,

Error Code = INVALID\_DATATYPE

'First Failed Element Number' = (the position in the 'List Of Initial Values' with the offending value)

ELSE

RECEIVE CreateObject-Error,

Error Class = PROPERTY,

Error Code = VALUE\_OUT\_OF\_RANGE | INVALID\_DATATYPE | OTHER

'First Failed Element Number' = (the position in the 'List Of Initial Values' with the offending value) |

(BACnet-Reject-PDU

Reject Reason = INVALID\_PARAMETER\_DATATYPE |

INVALID\_TAG)

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = (the 'Object Identifier' used in step 1),

'Property Identifier' = Object\_Name

1. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 4) THEN

RECEIVE BACnet-Error PDU,

Error Class = OBJECT,

Error Code = UNKNOWN\_OBJECT

ELSE

RECEIVE BACnet-Error PDU

Error Class = OBJECT,

Error Code = UNKNOWN\_OBJECT |

NO\_OBJECTS\_OF\_SPECIFIED\_TYPE | OTHER

1. 尝试创建实例为4194303的对象

目的：此测试用例验证当“对象说明符”参数传送具有4194303实例的对象标识符时，CreateObject服务请求的正确执行。仅当Device对象中存在Protocol\_Revision属性并且具有值大于或等于4执行。

测试步骤：

1. TRANSMIT CreateObject-Request,

'Object Identifier' = (any object identifier representing a creatable object-type with an instance of 4194303)

1. RECEIVE BACnet-Reject-PDU,

'Reject Reason' = PARAMETER\_OUT\_OF\_RANGE

* 1. 删除对象（DeleteObject）服务执行测试

本节定义了演示支持执行DeleteObject服务请求所必需的测试。

所需测试：ReadProperty服务执行测试，10.18。

BACnet参考条款：15.4。

在本节中定义的每个测试中，都有一个步骤，读入Device对象的Object\_List列表属性以验证新删除的对象不再出现在列表中。如果Object\_List很长，则此过程可能不适用于不支持分段的实现。在这些情况下，可通过读取Object\_List数组的每个元素，一次读取一个元素并验证新删除的对象不存在来代替。

1. 正响应DeleteObject服务执行测试

此测试组的目的是在预期服务成功完成的情况下验证DeleteObject服务请求的正确执行。

1. 成功删除对象

目的：验证成功删除对象的能力。

配置要求：IUT应配置一个可删除的对象X.

测试步骤：

1. VERIFY (X), Object\_Name = (any valid value)
2. TRANSMIT DeleteObject-Request,

'Object Identifier' = X

1. RECEIVE BACnet-Simple-ACK-PDU
2. TRANSMIT ReadProperty-Request,

'Object Identifier' = X,

'Property Identifier' = Object\_Name

1. RECEIVE BACnet-Error-PDU,

Error Class = OBJECT,

Error Code = UNKNOWN\_OBJECT

1. VERIFY (X), Object\_List = (any object list that does not contain X)
2. 负响应DeleteObject服务执行测试

此测试组的目的是在预期服务失败的情况下验证DeleteObject服务请求的正确执行。

1. 尝试删除不可删除的对象

目的：验证对尝试删除不可删除对象的正确响应。

配置要求：IUT应配置一个无法删除的对象X.

测试步骤：

1. TRANSMIT DeleteObject-Request,

'Object Identifier' = X

1. RECEIVE BACnet-Error-PDU,

Error Class = OBJECT,

Error Code = OBJECT\_DELETION\_NOT\_PERMITTED

1. VERIFY (X), Object\_Name = (the Object\_Name specified in the EPICS)
2. VERIFY (X), Object\_List = (any object list that contains X)
3. 尝试删除不存在的对象

目的：验证对尝试删除不存在的对象的正确响应。

测试步骤：

1. TRANSMIT DeleteObject-Request,

'Object Identifier' = X

1. RECEIVE BACnet-Error-PDU,

Error Class = OBJECT,

Error Code = UNKNOWN\_OBJECT

* 1. 读属性（ReadProperty）服务执行测试

本节定义了证明支持执行ReadProperty服务请求所必需的测试。

所需测试：无。

BACnet参考条款：15.5。

1. 正响应ReadProperty服务执行测试

此测试组的目的是在预期服务成功完成的情况下验证ReadProperty服务请求的正确执行。设X是IUT的设备对象的实例编号。

1. 读取数组的大小

目的：当请求的属性是数组并且请求数组的大小时，验证IUT是否可以执行ReadProperty服务请求。

测试步骤：

1. VERIFY (Device, X), Object\_List = (the size of the Object\_List specified in the EPICS), ARRAY INDEX = 0

1. 读取数组的单个元素

目的：当请求的属性是数组并且请求数组的单个元素时，验证IUT是否可以执行ReadProperty服务请求。

测试步骤：

1. VERIFY (Device, X),

Object\_List = (the first element of the Object\_List array as specified in the EPICS),

ARRAY INDEX = 1

1. 使用未知实例从设备对象中读取属性

目的：此测试用例验证当请求的对象标识符引用具有未知实例的设备对象时，IUT可以执行ReadProperty服务请求（4194303）。设X是IUT的设备对象的实例编号。仅当Device对象中存在Protocol\_Revision属性且值大于或等于4时，才应执行此测试。

测试步骤：

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = (Device, 4194303),

'Property Identifier' = Object-Identifier

1. RECEIVE ReadProperty-ACK,

'Object Identifier' = (Device, X),

'Property Identifier' = Object-Identifier,

'Property Value' = (Device, X)

传输结果：IUT应按照指示响应传送EPICS中指定的值。

1. 读取整个数组

目的：当请求的属性是数组并且请求内容为整个数组时，验证IUT是否可以执行ReadProperty服务请求。

BACnet参考条款：5.4.5.3

测试步骤：

1. VERIFY (Device, X), Object\_List = (the entire Object\_List array)

传送结果：IUT应按照指示响应，传送EPICS中指定的值。如果给定IPDU和TD的APDU和分段限制，如果对象列表太长而无法返回，则表明“不支持分段”或“缓冲区溢出”的中止消息会通过。如果接收到中止消息，并且IUT拥有另一个足够小的阵列，可以完整读取而不进行分段，则应使用该阵列重复此测试。在这种情况下，传送的结果是响应ReadProperty请求返回了整个数组。

1. 负响应ReadProperty服务执行测试

此测试组的目的是在预期服务失败的情况下验证ReadProperty服务请求的正确执行。

1. 使用数组索引读取非数组属性

目的：当请求的属性值不是数组但服务请求中包含数组索引时，验证IUT是否可以执行ReadProperty服务请求。

测试步骤：

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = (Device, X),

'Property Identifier' = Vendor\_Name,

'Array Index' = 1

1. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 4) THEN

RECEIVE BACnet-Error PDU,

Error Class = PROPERTY,

Error Code = PROPERTY\_IS\_NOT\_AN\_ARRAY

ELSE

RECEIVE

(BACnet-Error-PDU,

Error Class = PROPERTY,

Error Code = PROPERTY\_IS\_NOT\_AN\_ARRAY | INVALID\_ARRAY\_INDEX) |

(BACnet-Error-PDU,

Error Class = SERVICES,

Error Code = INCONSISTENT\_PARAMETERS)

1. 读取数组索引超出范围的数组属性

目的：当请求的属性值是数组但数组索引超出范围时，验证IUT是否可以执行ReadProperty服务请求。

测试步骤

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_List,

'Array Index' = (a value larger than the size of the Object\_List)

1. RECEIVE BACnet-Error-PDU,

Error Class = PROPERTY,

Error Code = INVALID\_ARRAY\_INDEX

1. 读取未知对象

目的：验证IUT是否可以在请求的对象不存在的情况下执行ReadProperty服务请求。

测试思想：TD尝试读取未为指定对象定义的属性。

测试步骤：

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = (any standard object not contained in the IUT's database),

'Property Identifier' = (any property defined for the specified object)

2. RECEIVE BACnet-Error-PDU,

Error Class = OBJECT,

Error Code = UNKNOWN\_OBJECT

1. 读取未知属性

目的：验证IUT是否可以在请求的属性不存在的情况下执行ReadProperty服务请求。

测试思想：TD尝试读取未为指定对象定义的属性。

测试步骤：

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = (Device, X),

'Property Identifier' = (any property defined for the specified object)

1. RECEIVE BACnet-Error-PDU,

Error Class = PROPERTY,

Error Code = UNKNOWN\_PROPERTY

* 1. （ReadPropertyConditional）服务执行测试

此部分定义了证明支持执行ReadPropertyConditional服务请求所必需的测试。

由于ReadPropertyConditional服务的灵活性和复杂性，以下小节包含测试要求而不是特定测试。 必须根据对IUT测试数据库的分析，为特定IUT定制这些测试的详细信息。 总之，测试应该涵盖广泛的情况：不同的布尔运算，不同的比较运算符，不同的属性值数据类型以及响应中返回的不同属性。所有BACnet对象都必须支持Object\_Identifier，Object\_Name和Object\_Type属性，因此这些属性可能是用于开发特定IUT的特定测试的便捷属性。

所需测试：无。

BACnet参考条款：15.6。

1. 对象数据库中有匹配项的“OR”选择逻辑

目的：验证IUT是否正确执行了使用“OR”选择逻辑的ReadPropertyConditional服务请求，并使与IUT数据库中的一个或多个对象匹配。

测试思想：'OR'选择逻辑应与两个或多个选择标准一起使用。标准选择应保证至少一个对象只满足一个选择标准。优选地，对于每个选择标准，应该存在一个或多个满足该特定标准的对象。在特定准则中使用的“Comparison Value”应与IUT数据库中至少一个对象的“Property Identifier”选择的属性具有相同的数据类型。“List of Property References”参数应包含一个或多个可能满足选择标准的对象中存在的属性。

测试步骤：TD应发送符合测试思想中所述要求的ReadPropertyConditional服务请求。

注意事项：IUT应使用正确编码的BACnet-Complex-ACK进行响应。 “List of Read Access Results”参数应包含限制为满足至少一个选择标准的对象的值列表，以及在请求的“属性引用列表”参数中指定的属性。

1. 'OR'否定测试

目的：验证IUT是否正确执行了使用“OR”选择逻辑的ReadPropertyConditional服务请求，并使与IUT数据库中没有对象匹配。

测试思想：“OR”选择逻辑应与两个或多个选择标准一起使用。 选择标准应保证没有任何对象满足任何选择标准。对于每个选择标准，应该有一个或多个对象包含该标准中使用的属性。特定标准中使用的“Comparison Value”应与IUT测试数据库中至少一个对象的“Property Identifier”选择的属性具有相同的数据类型。“List of Property References”参数应包含一个或多个任意属性。

测试步骤：TD应传输符合测试思想要求的ReadPropertyConditional服务请求。

注意事项：IUT应使用正确编码的BACnet-Complex-ACK进行响应。 “List of Read Access Results”参数应为空（零长度）列表。

* 1. 读多个属性（ReadPropertyMultiple）服务执行测试

本节定义了证明支持执行ReadPropertyMultiple服务请求所必需的测试。

所需测试：无。

BACnet参考条款：15.7。

配置要求：IUT应在其数据库中配置至少两个BACnet对象。

测试思想：测试人员应从IUT的数据库中选择两个对象。各种测试包括从这些对象中的一个或两个读取属性组合。 在测试描述中，这些对象的Object\_Identifier分别指定为Object1和Object2。测试人员选择的属性根据需要指定为P1，P2，P3等。

1. 正响应ReadPropertyMultiple服务执行测试

此测试组的目的是在预期服务成功完成的情况下验证ReadPropertyMultiple服务请求的正确执行。

1. 从单个对象读取单个属性

目的：验证从单个对象读取单个属性的能力。

测试思想：从Device对象读取单个受支持的属性。该属性由TD选择，并在测试描述中指定为P1。

测试步骤：

1. TRANSMIT ReadPropertyMultiple-Request,

'Object Identifier' = Object1 | Object2,

'Property Identifier' = P1

1. RECEIVE ReadPropertyMultiple-ACK,

'Object Identifier' = (the object selected in step 1),

'Property Identifier' = P1,

'Property Value' = (the value of P1 specified in the EPICS)

1. 从单个对象中读取多个属性

目的：验证从单个对象读取多个属性的能力。

测试步骤

1. TRANSMIT ReadPropertyMultiple-Request,

'Object Identifier' = Object1 | Object 2,

'Property Identifier' = P1,

'Property Identifier' = P2

-- … (Two properties are required but more may be selected.)

1. RECEIVE ReadPropertyMultiple-ACK,

'Object Identifier' = (the object selected in step 1),

'Property Identifier' = P1,

'Property Value' = (the value of P1 specified in the EPICS),

'Property Identifier' = P2,

'Property Value' = (the value of P2 specified in the EPICS)

-- … (An appropriate value must be returned for each property included in the ReadPropertyMultiple-Request.)

1. 从多个对象中读取单个属性

目的：验证从多个对象读取单个属性的能力。

测试步骤：

1. TRANSMIT ReadPropertyMultiple-Request,

'Object Identifier' = Object1,

'Property Identifier' = P1,

'Object Identifier' = Object2,

'Property Identifier' = P2

-- … (Two properties are required but more may be selected.)

1. RECEIVE ReadPropertyMultiple-ACK,

'Object Identifier' = Object1,

'Property Identifier' = P1,

'Property Value' = (the value of P1 specified in the EPICS),

'Object Identifier' = Object2,

'Property Identifier' = P2,

'Property Value' = (the value of P2 specified in the EPICS)

-- … (An appropriate value must be returned for each property included in the

ReadPropertyMultiple-Request.)

1. 从多个对象中读取多个属性

目的：验证从多个对象读取多个属性的能力。

测试步骤：

1. TRANSMIT ReadPropertyMultiple-Request,

'Object Identifier' = Object1,

'Property Identifier' = P1,

'Property Identifier' = P2,

'Property Identifier' = P3,

'Object Identifier' = Object2,

'Property Identifier' = P4,

'Property Identifier' = P5,

'Property Identifier' = P6

-- … (Two objects must be included but but more may be selected.)

1. RECEIVE ReadPropertyMultiple-ACK,

'Object Identifier' = Object1,

'Property Identifier' = P1,

'Property Value' = (the value of P1 specified in the EPICS),

'Property Identifier' = P2,

'Property Value' = (the value of P2 specified in the EPICS),

'Property Identifier' = P3,

'Property Value' = (the value of P3 specified in the EPICS),

'Object Identifier' = Object2,

'Property Identifier' = P4,

'Property Value' = the value of P4 specified in the EPICS)

'Property Identifier' = P5,

'Property Value' = (the value of P5 specified in the EPICS),

'Property Identifier' = P6

'Property Value' = (the value of P6 specified in the EPICS)

-- … (An appropriate value must be returned for each property included in the

ReadPropertyMultiple-Request.)

1. 读取具有单个嵌入式访问错误的多个属性

目的：验证是否能够正确执行ReadPropertyMultiple服务请求，其中“读取访问标准列表”包含不受支持的属性的标准。

测试步骤：

1. TRANSMIT ReadPropertyMultiple-Request,

'Object Identifier' = Object1,

'Property Identifier' = P1,

'Property Identifier' = P2,

'Property Identifier' = (any property, P3, not supported in this object),

'Property Identifier' = P4

1. RECEIVE ReadPropertyMultiple-ACK,

'Object Identifier' = Object1,

'Property Identifier' = P1,

'Property Value' = (the value of P1 specified in the EPICS),

'Property Identifier' = P2,

'Property Value' = (the value of P2 specified in the EPICS),

'Property Identifier' = P3,

'Error Class' = PROPERTY,

'Error Code' = UNKNOWN\_PROPERTY,

'Property Identifier' = P4,

'Property Value' = (the value of P4 specified in the EPICS)

1. 读取多个嵌入式访问错误的多个属性

目的：验证是否能够正确执行ReadPropertyMultiple服务请求，其中“读取访问标准列表”包含多个不受支持的属性的标准。

测试步骤：

1. TRANSMIT ReadPropertyMultiple-Request,

'Object Identifier' = Object1,

'Property Identifier' = P1,

'Property Identifier' = P2,

'Property Identifier' = (any property, P3, not supported in this object),

'Property Identifier' = (any property, P4, not supported in this object),

'Object Identifier' = (any object, Object2, not supported in the IUT)

'Property Identifier' = P5,

'Property Identifier' = P6

1. RECEIVE ReadPropertyMultiple-ACK,

'Object Identifier' = Object1,

'Property Identifier' = P1,

'Property Value' = (the value of P1 specified in the EPICS),

'Property Identifier' = P2,

'Property Value' = (the value of P2 specified in the EPICS),

'Property Identifier' = P3,

'Error Class' = PROPERTY,

'Error Code' = UNKNOWN\_PROPERTY,

'Property Identifier' = P4,

'Error Class' = PROPERTY,

'Error Code' = UNKNOWN\_PROPERTY,

'Object Identifier' = Object2,

'Property Identifier' = P5,

'Error Class' = OBJECT,

'Error Code' = UNKNOWN\_OBJECT,

'Property Identifier' = P6,

'Error Class' = OBJECT,

'Error Code' = UNKNOWN\_OBJECT

1. 读取所有属性

目的：验证是否能够正确执行使用特殊属性标识符ALL的ReadPropertyMultiple服务请求。测试支持的每个对象类型的一个实例。

测试步骤：

1. REPEAT ObjectX = (one instance of each supported object type) DO {

TRANSMIT ReadPropertyMultiple-Request,

'Object Identifier' = ObjectX,

'Property Identifier' = ALL

RECEIVE ReadPropertyMultiple-ACK,

'Object Identifier' = Object1,

REPEAT P = (each property supported by Object1) DO {

'Property Identifier' = P,

'Property Value' = (the value of P specified in the EPICS)

}

}

测试人员注意事项：对象类型支持的任何专有属性也应返回（参见BACnet 15.7.3.1.2）。

1. 读取可选属性

目的：验证是否能够正确执行使用特殊属性标识符OPTIONAL的ReadPropertyMultiple服务请求。测试支持的每个对象类型的一个实例。

测试步骤：

1. REPEAT ObjectX = (one instance of each supported object type) DO {

TRANSMIT ReadPropertyMultiple-Request,

'Object Identifier' = Object1,

'Property Identifier' = OPTIONAL

RECEIVE ReadPropertyMultiple-ACK,

'Object Identifier' = Object1,

REPEAT P = (each optional property supported by Object1) DO {

'Property Identifier' = P,

'Property Value' = (the value of P specified in the EPICS)

}

}

测试人员注意事项：如果不支持可选属性，则应为指定属性返回空的“List of Results”。

1. 读取所需属性

目的：验证是否能够正确执行使用特殊属性标识符REQUIRED的ReadPropertyMultiple服务请求。测试支持每个对象类型的一个实例。

测试步骤：

1. REPEAT ObjectX = (one instance of each supported object type) DO {

TRANSMIT ReadPropertyMultiple-Request,

'Object Identifier' = Object1,

'Property Identifier' = REQUIRED

RECEIVE ReadPropertyMultiple-ACK,

'Object Identifier' = Object1,

REPEAT P = (each required property defined for Object1) DO {

'Property Identifier' = P,

'Property Value' = (the value of P specified in the EPICS)

}

}

1. 读取数组的大小

目的：当请求的属性是数组并且请求数组的大小时，验证IUT是否可以执行ReadPropertyMultiple服务请求。

测试思想：使用ReadPropertyMultiple而不是ReadProperty重复10.18.1.1中的测试，读取数组的大小。

1. 使用未知实例从设备对象中读取属性

目的：此测试案例验证当所请求的对象标识符引用具有未知实例的设备对象时，IUT可以执行ReadPropertyMultiple服务请求（4194303）。设X是IUT的设备对象的实例编号。 仅当Device对象中存在Protocol\_Revision属性且值大于或等于4时，才执行此测试。

测试步骤：

1. TRANSMIT ReadPropertyMultiple-Request,

'Object Identifier' = (Device, 4194303),

'Property Identifier' = Object-Identifier

1. RECEIVE ReadPropertyMultiple-ACK,

'Object Identifier' = (Device, X),

'Property Identifier' = Object-Identifier,

'Property Value' = (Device, X)

传输结果：IUT必须按照指示进行响应，传达EPICS中指定的值。

1. 读取最大多个属性

目的：此测试用例验证IUT不会任意限制可以使用单个ReadPropertyMultiple请求读取的属性的数量。

测试思想：从设备对象中读取对象标识符的次数可以在IUT接受的最大请求中传送，或者可以在IUT可以生成的最大响应中返回。最大请求/响应大小的计算应基于IUT的Max\_APDU\_Length\_Accepted和每个请求/响应的最大段。

确定要使用的对象标识符数量的过程是：

MaxAPDU = IUT's Max\_APDU\_Length\_Accepted

MaxRxSegs = IUT's maximum segments accepted per request

MaxTxSegs = IUT's maximum segments generated per response

NonSegRqstHdrSize = size of (non-segmented BACnetConfirmed-RequestPDU header) = 4

SegRqstHdrSize = size of (segmented BACnetConfirmed-RequestPDU header) = 6

NonSegRespHdrSize = size of (non-segmented BACnet-ComplexACK-PDU header) = 3

SegRespHdrSize = size of (segmented BACnet-ComplexACK-PDU header) = 5

ObjIdSize = size of (an Object-Identifier) = 5

TagsSize = size of (an open and a close tag) = 2

PropIdSize = size of ('Object-Identifier' property Id) = 2

如果IUT不支持接收分段请求:

MaxPropsPerRqst =

(MaxAPDU - NonSegRqstHdrSize - ObjIdSize - TagsSize) / PropIdSize =

(MaxAPDU - 11) / 2

如果IUT支持接收分段请求:

MaxPropsPerRqst =

((MaxAPDU - SegRqstHdrSize) \* MaxRxSegs - ObjIdSize - TagsSize) / PropIdSize =

((MaxAPDU - 6) \* MaxRxSegs - 7) / 2

如果IUT支持发送分段响应:

MaxPropsPerResp =

(MaxAPDU - NonSegRespHdrSize - ObjIdSize - TagsSize) / (PropIdSize + TagsSize + ObjIdSize) = (MaxAPDU - 10) / 9

如果IUT支持发送分段响应:

MaxPropsPerResp =

((MaxAPDU - SegRespHdrSize) \* MaxTxSegs - ObjIdSize - TagsSize) / (PropIdSize +

TagsSize + ObjIdSize) = ((MaxAPDU - 5) \* MaxTxSegs - 7) / 9

NumPropertiesToUse = min(MaxPropsPerRqst, MaxPropsPerResp)

测试步骤：

1. TRANSMIT ReadPropertyMultiple-Request,

'Object Identifier' = (Device, X),

'Property Identifier' = Object-Identifier,

'Property Identifier' = Object-Identifier,

'Property Identifier' = Object-Identifier,

…

'Property Identifier' = Object-Identifier

1. RECEIVE ReadPropertyMultiple-ACK,

'Object Identifier' = (Device ,X),

'Property Identifier' = Object-Identifier,

'Property Value' = (Device, X),

'Property Identifier' = Object-Identifier,

'Property Value' = (Device, X),

'Property Identifier' = Object-Identifier,

'Property Value' = (Device, X),

…

'Property Identifier' = Object-Identifier,

'Property Value' = (Device, X)

1. 负响应ReadPropertyMultiple服务执行测试

此测试目的是在预期服务失败的情况下验证ReadPropertyMultiple服务请求的正确执行。

1. 从单个对象读取单个不受支持的属性

目的：验证是否能够正确执行ReadPropertyMultiple服务请求，其中“读取访问标准列表”包含单个不受支持的属性的标准。

测试步骤：

1. TRANSMIT ReadPropertyMultiple-Request,

'Object Identifier' = Object1 | Object2,

'Property Identifier' = (any property, P1, that is not supported in the selected object)

1. RECEIVE BACnet-Error-PDU,

'Error Class' = PROPERTY,

'Error Code' = UNKNOWN\_PROPERTY

| ( ReadPropertyMultiple-ACK,

'Object Identifier' = (the object identifier from step 1),

'Property Identifier' = P1,

'Error Class' = PROPERTY,

'Error Code' = UNKNOWN\_PROPERTY)

1. 读取具有每个属性访问错误的多个属性

目的：验证是否能够正确执行ReadPropertyMultiple服务请求，其中“读取访问标准列表”仅包含不受支持的属性的标准。

测试思想：此测试的对象和属性的选择应由不支持的对象、所选对象不支持的属性或两者的组合（所以没有对象）组成代表支持的属性的属性组合。

测试步骤：

1. TRANSMIT ReadPropertyMultiple-Request,

'Object Identifier' = Object1,

'Property Identifier' = P1,

'Property Identifier' = P2,

'Property Identifier' = P3,

'Object Identifier' = Object2,

'Property Identifier' = P4,

'Property Identifier' = P5,

'Property Identifier' = P6

1. RECEIVE

(BACnet-Error-PDU,

'Error Class' = OBJECT | PROPERTY,

'Error Code' = (any valid error code for the returned error class) )|

(ReadPropertyMultiple-ACK,

'Object Identifier' = Object1,

'Property Identifier' = P1,

'Error Class' = OBJECT | PROPERTY,

'Error Code' = (any valid error code for the returned error class),

'Property Identifier' = P2,

'Error Class' = OBJECT | PROPERTY,

'Error Code' = (any valid error code for the returned error class),

'Property Identifier' = P3,

'Error Class' = OBJECT | PROPERTY,

'Error Code' = (any valid error code for the returned error class),

'Object Identifier' = Object2,

'Property Identifier' = P4,

'Error Class' = OBJECT | PROPERTY,

'Error Code' = (any valid error code for the returned error class),

'Property Identifier' = P5,

'Error Class' = OBJECT | PROPERTY,

'Error Code' = (any valid error code for the returned error class),

'Property Identifier' = P6,

'Error Class' = OBJECT | PROPERTY,

'Error Code' = (any valid error code for the returned error class) )

1. 使用数组索引读取单个非数组属性

目的：此测试用例验证当请求的属性值不是数组但服务请求中包含数组索引时，IUT可执行ReadPropertyMultiple服务请求。仅当Protocol\_Revision存在且值大于或等于4时，才应执行此测试。

测试步骤：

1. TRANSMIT ReadPropertyMultiple-Request,

'Object Identifier' = (Device, X),

'Property Identifier' = Vendor\_Name,

'Array Index' = 1

1. RECEIVE

(BACnet-Error-PDU,

'Error Class' = PROPERTY,

'Error Code' = PROPERTY\_IS\_NOT\_AN\_ARRAY) |

(ReadPropertyMultiple-ACK,

'Object Identifier' = (Device, X),

'Property Identifier' = Vendor\_Name,

'Array Index'= 1,

'Error Class'= PROPERTY,

'Error Code' = PROPERTY\_IS\_NOT\_AN\_ARRAY)

* 1. 读范围（ReadRange）服务执行测试

本节定义了证明支持执行ReadRange服务请求所必需的测试。

所需测试：无。

BACnet参考条款：15.8。

配置要求：IUT必须配置有包含一组已知趋势数据的趋势日志对象。TD必须具有趋势数据的确切内容，以便评估测试结果。Log\_Enable属性的值应为FALSE，以便在测试期间Log\_Buffer不变。

以下样本缓冲区用作本节中测试的说明。

**表10-1 样本Log\_Buffer, (趋势日志，实例1)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **任意记录指定** | **位置(索引**) | **隐含序列#** | **时间戳**  **(为清晰起见，不包括日期)** | **日志数据** |
| a  b  c  d  e  f  g  h  i  g  k | 1  2  3  4  5  6  7  8  9  10  11 | 16  17  18  19  20  21  22  23  24  25  26 | 13:01:00:00  13:02:00:00  13:05:00:00  13:10:00:00  13:15:00:00  13:16:00:00  13:21:00:00  13:25:00:00  13:30:00:00  13:35:00:00  13:36:00:00 | log-status, buffer-purged  log-status, log-disabled = FALSE  real-value = 5.0  real-value = 10.0  real-value = 15.0  log-status, log-disabled = TRUE  log-status, log-disabled = FALSE  real-value = 25.0  real-value = 30.0  real-value = 35.0  log-status, log-disabled = TRUE |

1. 正响应ReadRange服务执行测试

此测试组的目的是在预期服务成功完成的情况下验证ReadRange服务请求的正确执行。

1. 读取列表中的所有项目

目的：验证IUT是否正确响应ReadRange服务请求以返回所有可用数据项。

测试步骤：

1. TRANSMIT ReadPropertyMultiple-Request

'Object Identifier' = (the log object configured for this test),

'Property Identifier' = Log\_Buffer

1. RECEIVE Read-Range-ACK,

'Object Identifier' = (the log object configured for this test),

'Property Identifier' = Log\_Buffer,

'Result Flags' = {TRUE, TRUE, FALSE},

'Item Count' = (the number of trend records in the test object),

'Item Data' = (all of the trend records in the test object)

注意事项：趋势数据可能包含的项目数多于单个消息中可返回的项目数。在这些情况下，“Result Flags”值为{TRUE，FALSE，TRUE}，“Item Count”和“Item Data”参数将反映能够返回的实际项目数。

1. 按正计数位置读取项目

目的：验证IUT是否正确响应ReadRange服务请求，以通过指示位置和返回该位置之后的项目数来返回指定的项目。

测试思想：TD发送ReadRange请求，请求在Log\_Buffer已知的一系列项目。使用“By Position”选项和“Count”的正值可以指定此范围。选择“Reference Index”和“Count”，以便可以在单个确认中传送结果。

测试步骤：

1. TRANSMIT ReadRange-Request,

'Object Identifier' = (the log object configured for this test),

'Property Identifier' = Log\_Buffer,

'Reference Index' = (any value x: 1 ≤ x ≤ Record\_Count),

'Count' = (any value x: 0 < x ≤ Record\_Count - x + 1)

1. RECEIVE Read-Range-ACK,

'Object Identifier' = (the log object configured for this test),

'Property Identifier' = Log\_Buffer,

'Result Flags' = {?, ?, FALSE},

'Item Count' = (the same value used in the 'Count' parameter in step 1),

'Item Data' = (all of the specified trend records in order of increasing position. The items specified include the item at the index specified by x, plus (y-1) items following.)

测试示例（使用部分开头的样本缓冲区）：

1. TRANSMIT ReadRange-Request,

'Object Identifier' = (Trend Log, Instance 1),

'Property Identifier' = Log\_Buffer,

'Reference Index' = 3,

'Count' = 7

1. RECEIVE ReadRange-ACK,

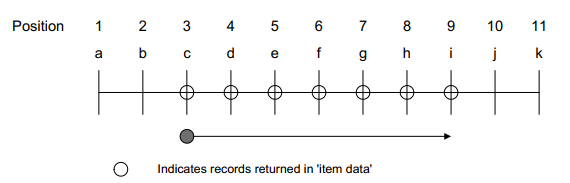
'Object Identifier' = (Trend Log, Instance 1),

'Property Identifier' = Log\_Buffer,

'Result Flags' = {FALSE, FALSE, FALSE},

'Item Count' = 7,

'Item Data' = Records < c, d, e, f, g, h, i > in that order.



**图10-1** 按正计数位置读取项目

1. 按负计数位置读取项目

目的：验证IUT是否正确响应ReadRange服务请求，以通过指示位置和返回该位置之前的项目数来返回指定的项目。

测试思想：TD发送ReadRange请求，请求在Log\_Buffer中已知的一系列项目。使用“By Position”选项和“Count”为负值来指定此范围。选择“Reference Index”和“Count”，以便可以在一次确认中传达结果。

测试步骤：

1. TRANSMIT ReadRange-Request,

'Object Identifier' = (the log object configured for this test),

'Property Identifier' = Log\_Buffer,

'Reference Index' = (any value x: 1 ≤ x ≤ Record\_Count),

'Count' = (any value y: y < 0 AND |y| ≤ x)

1. RECEIVE ReadRange-ACK,

'Object Identifier' = (the log object configured for this test),

'Property Identifier' = Log\_Buffer,

'Result Flags' = {?, ?, FALSE},

'Item Count' = |y|,

'Item Data' = (all of the specified trend records in order of increasing position. The items specified include the item at the index specified by x, plus |y|-1 items preceding.)

测试示例（使用部分开头的样本缓冲区）：

1. TRANSMIT ReadRange-Request,

'Object Identifier' = (Trend Log, Instance 1),

'Property Identifier' = Log\_Buffer,

'Reference Index' = 8,

'Count' = -8

1. RECEIVE ReadRange-ACK,

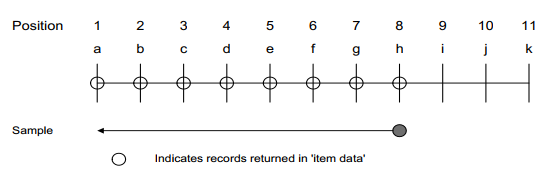
'Object Identifier' = (Trend Log, Instance 1),

'Property Identifier' = Log\_Buffer,

'Result Flags' = {TRUE, FALSE, FALSE},

'Item Count' = 8

'Item Data' = Records < a, b, c, d, e, f, g, h > in that order.



**图10-2** 按负计数位置读取项目

1. 按时间读取项目

目的：验证IUT是否正确响应ReadRange服务请求，以通过指示时间和返回之后的项目数来返回指定的项目。

测试思想：TD发送ReadRange请求，请求在Log\_Buffer中已知的一系列项目。使用“By Time”选项和“Count”的正值可以指定此范围。选择“'Reference Index”和“Count'”，以便可以在一次确认中传送结果。

测试步骤：

1. TRANSMIT ReadRange-Request,

'Object Identifier' = (the log object configured for this test),

'Property Identifier' = Log\_Buffer,

'Reference Time' = (any value x: x is older (of earlier time) than the time of an entry in the buffer which has a sequence number of S, and x is newer than or equal to the time of any preceding entry),

'Count' = (any value y: 0 < y ≤ Total\_Record\_Count – S+ 1)

1. RECEIVE ReadRange-ACK,

'Object Identifier' = (the log object configured for this test),

'Property Identifier' = Log\_Buffer,

'Result Flags' = {?, ?, FALSE},

'Item Count' = y,

'Item Data' = (all of the specified trend records in order of increasing sequence number. The items specified include the first item with a timestamp newer than x, plus (y-1) items following.)

'First Sequence Number' = S

测试示例（在部分开头使用样本缓冲区）：

1. TRANSMIT ReadRange-Request,

'Object Identifier' = (Trend Log, Instance 1),

'Property Identifier' = Log\_Buffer,

'Reference Time' = 13:21:00.00

'Count' = 4

1. RECEIVE ReadRange-ACK,

'Object Identifier' = (Trend Log, Instance 1),

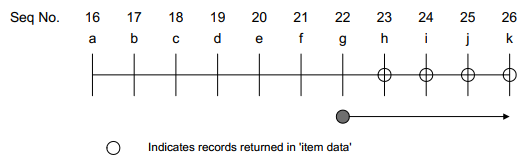
'Property Identifier' = Log\_Buffer,

'Result Flags' = {FALSE, TRUE, FALSE},

'Item Count' = 4,

'Item Data' = Records < h, i, j, k > in that order.

'First Sequence Number'= 23



**图10-3** 按时间读取项目

1. 用负计数按时间读取项目

目的：验证IUT是否正确响应ReadRange服务请求，以通过指示时间和返回之后的项目数来返回指定的项目。

测试思想：TD发送ReadRange请求，请求在Log\_Buffer中已知的一系列项目。使用“By Time”选项和“Count”为负值来指定此范围。选择的“Reference Time”x应该比缓冲区中的上次更新。选择“Reference Time”和“Count”，以便可以在一次确认中传送结果。

测试配置：配置TD，使得不发生时间变化请求。 配置TD使其在Log\_Buffer中至少包含3个项目。

测试步骤：

1. TRANSMIT ReadRange-Request,

'Object Identifier' = (the log object configured for this test),

'Property Identifier' = Log\_Buffer,

'Reference Time' = (x, selected as described above),

'Count' = (any value y: 0 < |y| ≤ number of records in the buffer)

1. RECEIVE ReadRange-ACK,

'Object Identifier' = (the log object configured for this test),

'Property Identifier' = Log\_Buffer,

'Result Flags' = {?, TRUE, FALSE},

'Item Count' = |y|,

'Item Data' = (All of the specified trend records in order of increasing sequence number. The items specified include the last item with a timestamp older than x, plus |y|-1 items preceding.)

'First Sequence Number' = (Total\_Record\_Count- |y| + 1)

注意事项：返回的所有项目都应包含一个早于参考时间参数指定时间的时间戳。返回的项目应该是日志缓冲区中的最后一个“Count”项。 如果Log\_Buffer中有一个条目，其时间戳与“Reference Time”参数完全匹配，则该条目不应包含在“Item Data”中。

测试示例（使用部分开头的样本缓冲区）：

1. TRANSMIT ReadRange-Request,

'Object Identifier' = (Trend Log, Instance 1),

'Property Identifier' = Log\_Buffer,

'Reference Time' = 13:40:00.00,

'Count' = -10

1. RECEIVE ReadRange-ACK,

'Object Identifier' = (Trend Log, Instance 1),

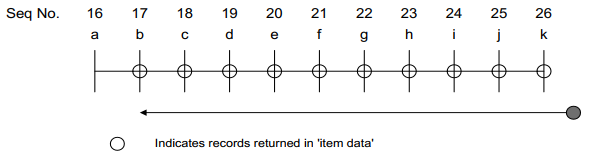
'Property Identifier' = Log\_Buffer,

'Result Flags' = {FALSE, TRUE, FALSE},

'Item Count' = 10,

'Item Data' = (records < b, c, d, e, f, g, h, i, j, k > in that order.)

'First Sequence Number' = 17



**图10-4** 用负计数按时间读取项目

1. 按时间范围读取项目

目的：验证IUT是否正确响应ReadRange服务请求，以通过指示要包括的时间范围来返回指定的项目。

测试思想：TD发送ReadRange请求，请求在Log\_Buffer中已知的一系列项目。使用“Time Range”选项指定此范围。 选择“Beginning Time”和“Ending Time”，以便可以在单个确认中传送结果。

测试步骤：

1. TRANSMIT ReadRange-Request,

'Object Identifier' = (the Trend Log object configured for this test),

'Property Identifier' = Log\_Buffer,

'Beginning Time' = (any value before the last time in the buffer),

'Ending Time' = (any value > 'Beginning Time')

1. RECEIVE Read-Range-ACK,

'Object Identifier' = (the Trend Log object configured for this test),

'Property Identifier' = Log\_Buffer,

'Result flags' = {TRUE, TRUE, FALSE},

'Item Count' = (the number of trend records meeting the specified criteria),

'Item Data' = (all of the specified trend records)

注意事项：返回的第一个项应该是缓冲区中第一个具有比“Beginning Time”参数指定的时间更新（更晚的时间）的时间戳。返回的最后一项应该是时间戳较早（较早的时间）大于或等于“Ending Time”参数指定的时间戳的项目

1. 读取一系列不存在的项目

目的：当没有指定范围内的项目时，验证IUT是否正确响应了ReadRange服务请求。

测试思想：TD发送ReadRange请求，请求一系列已知不在Log\_Buffer中的项目。 IUT应通过返回一个空列表来响应。

测试步骤：

1. TRANSMIT ReadRange-Request,

'Object Identifier' = (the Trend Log object configured for this test),

'Property Identifier' = Log\_Buffer,

'Beginning Time' = (any value that will result in a time interval for

which there are no items present),

'Ending Time' = (any value that will result in a time interval for

which there are no items present)

1. RECEIVE Read-Range-ACK,

'Object Identifier' = (the Trend Log object configured for this test),

'Property Identifier' = Log\_Buffer,

'Result flags' = {TRUE, TRUE, FALSE},

'Item Count' = 0,

'Item Data' = (an empty list)

1. 读取一系列不存在的项目（按顺序使用）

目的：当指定条件中没有项目时，验证IUT是否正确响应了ReadRange服务请求。

测试思想：TD发送ReadRange请求，请求指定的序列号和已知不在Log\_Buffer中的项目数。IUT应通过返回一个空列表来响应。

测试步骤：

1. TRANSMIT ReadRange-Request,

'Object Identifier' = (the log object configured for this test),

'Property Identifier' = Log\_Buffer,

'Reference Sequence Number' = (any value that will result in no items being present)

'Count' = (any non-zero number)

1. RECEIVE ReadRange-ACK,

'Object Identifier' = (the log object configured for this test),

'Property Identifier' = Log\_Buffer,

'Result flags' = {FALSE, FALSE, FALSE},

'Item Count' = 0,

'Item Data' = (an empty list)

'First Sequence Number'= (should be absent)

测试示例（在部分开头使用样本缓冲区）：

1. TRANSMIT ReadRange-Request,

'Object Identifier' = (Trend Log, Instance 1),

'Property Identifier' = Log\_Buffer,

'Reference Sequence Number' = 34

'Count' = 4

1. RECEIVE ReadRange-ACK,

'Object Identifier' = (Trend Log, Instance 1),

'Property Identifier' = Log\_Buffer,

'Result flags' = {FALSE, FALSE, FALSE},

'Item Count' = 0,

'Item Data' = (an empty list)

1. 读取一系列不存在的项目（按时间使用）

目的：当指定条件中没有项目时，验证IUT是否正确响应了ReadRange服务请求。

测试思想：TD发送ReadRange请求，请求指定的参考时间和已知不在Log\_Buffer中的项目数。IUT应通过返回一个空列表来响应。

测试步骤：

1. TRANSMIT ReadRange-Request,

'Object Identifier' = (the log object configured for this test),

'Property Identifier' = Log\_Buffer,

'Reference Time' = (any value that will result in no items being present)

'Count' = (any non-zero number)

1. RECEIVE ReadRange-ACK,

'Object Identifier' = (the log object configured for this test),

'Property Identifier' = Log\_Buffer,

'Result flags' = {FALSE, FALSE, FALSE},

'Item Count' = 0,

'Item Data' = (an empty list)

'First Sequence Number' = (should be absent)

测试示例（在部分开头使用样本缓冲区）：

1. TRANSMIT ReadRange-Request,

'Object Identifier' = (Trend Log, Instance 1),

'Property Identifier' = Log\_Buffer,

'Reference Time' = 12:00:00.00

'Count' = -10

1. RECEIVE ReadRange-ACK,

'Object Identifier' = (Trend Log, Instance 1),

'Property Identifier' = Log\_Buffer,

'Result flags' = {FALSE, FALSE, FALSE},

'Item Count' = 0,

'Item Data' = (an empty list)

1. 按正计数顺序读取项目

目的：验证IUT是否正确响应ReadRange服务请求，以通过指示序列号和返回该序列之后的项目数来返回指定的项目。

测试思想：TD发送ReadRange请求，请求已知在Log\_Buffer中的一系列项目。此范围使用By Sequence选项指定，并为Count指定正值。选择“Reference Sequence Number”和“Count”，以便可以在单个确认中传达结果。

测试步骤：

1. TRANSMIT ReadRange-Request,

'Object Identifier' = (the log object configured for this test),

'Property Identifier' = Log\_Buffer,

'Reference Sequence Number' = (any value x: (Total\_Record\_Count – Record\_Count + 1) ≤ x ≤ (Total\_Record\_Count – y + 1)),

'Count' = (any value y: 0 < y ≤ Record\_Count)

1. RECEIVE ReadRange-ACK,

'Object Identifier' = (the log object configured for this test),

'Property Identifier' = Log\_Buffer,

'Result Flags' = {?, ?, FALSE},

'Item Count' = y,

'Item Data' = (All of the specified trend records in the order of increasing

sequence number. The items specified are all items with the

sequence number in the range of x through (x+y-1) in that order).

'First Sequence Number' = x

测试示例（在部分开头使用样本缓冲区）：

1. TRANSMIT ReadRange-Request,

'Object Identifier' = 20:1,

'Property Identifier' = Log\_Buffer,

'Reference Sequence Number' = 16,

'Count' = 11

1. RECEIVE ReadRange-ACK,

'Object Identifier' = 20:1,

'Property Identifier' = Log\_Buffer,

'Result Flags' = {TRUE, TRUE, FALSE},

'Item Count' = 11,

'Item Data' = Records < a, b, c, d, e, f, g, h, i, j, k > in that order.

'First Sequence Number' = 16

1. 按负计数顺序读取项目

目的：验证IUT是否正确响应ReadRange服务请求，以通过指示序列号和返回该序列之后的项目数来返回指定的项目。

测试思想：TD发送ReadRange请求，请求已知在Log\_Buffer中的一系列项目。

此范围使用By Sequence选项指定，负数值指定为Count。选择“Reference Sequence Number”和“Count”，以便可以在单个确认中传达结果。

测试步骤：

1. TRANSMIT ReadRange-Request,

'Object Identifier' = (the log object configured for this test),

'Property Identifier' = Log\_Buffer,

'Reference Sequence Number' = (any value x: (Total\_Record\_Count - Record\_Count+2) < x ≤ Record\_Count),

'Count' = (any value y: 0 < |y| < (Record\_Count - (Total\_Record\_Count – x) + 1)

1. RECEIVE ReadRange-ACK,

'Object Identifier' = (Trend Log, 1),

'Property Identifier' = Log\_Buffer,

'Result Flags' = {?, ?, FALSE},

'Item Count' = y,

'Item Data' = Records < d, e, f, g, h, i > in that order.

'First Sequence Number' = ( x - |y| + 1 )

测试示例（在部分开头使用样本缓冲区）：

1. TRANSMIT ReadRange-Request,

'Object Identifier' = (Trend Log, 1),

'Property Identifier' = Log\_Buffer,

'Reference Sequence Number' = 24,

'Count' = -6

1. RECEIVE ReadRange-ACK,

'Object Identifier' = (Trend Log, 1),

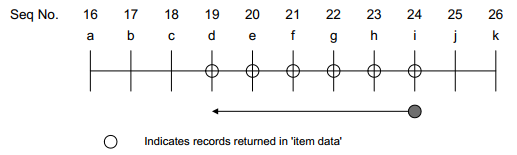
'Property Identifier' = Log\_Buffer,

'Result Flags' = {FALSE, FALSE, FALSE},

'Item Count' = 6,

'Item Data' = Records < d, e, f, g, h, i > in that order.

'First Sequence Number' = 19



**图10-5** 按负计数顺序读取项目

1. 数据类型验证测试

目的：验证ReadRange服务是否针对支持的每种数据类型正确执行。

测试思想：将Log\_DeviceObjectProperty设置为外部属性，并验证ReadRange在获取数据后是否正常执行。

配置要求：将Log\_DeviceObjectProperty设置为类型（REAL，UNSIGNED，INTEGER，BOOLEAN，BIT STRING，ENUMERATED）的外部属性。使IUT收集样本。然后，使用符合条件的9.21.1条测试之一来验证ReadRange-Request和ReadRange-Ack的操作。符合的测试为：10.2.1.1.1、10.2.11.2、10.2.1.3、10.2.1.4、10.21.1.4.1、10.2.11.7和10.21.1.8。

1. 状态/故障记录

目的：验证在尝试从受监视对象读取数据值时遇到错误时是否记录了故障。如果错误是通过远程设备的错误响应传达的，请验证是否记录了响应中的错误类和错误代码。

测试思想：通过将Log\_DeviceObjectProperty设置为无效的设备或对象，使受监视对象失败并响应错误。等到IUT尝试读取Log\_Buffer的样本。然后检查Log\_Buffer以验证是否存在由错误的ErrorClass和ErrorCode组成的失败条目。

测试步骤：

1. WRITE (Invalid object into the Log\_DeviceObjectProperty of the log object)

2. WAIT (until IUT attempts to read a sample for the Log\_Buffer)

3. VERIFY (Log\_Buffer contains a failure entry of unknown object)

* 1. 写属性(WriteProperty)服务执行测试

本节定义了验证支持执行WriteProperty服务请求的必需测试项。

所需测试：无。

BACnet参考条款：15.9。

测试思想：测试人员应从IUT数据库中选择一个适合的具有可写属性的对象作为测试用例。在测试描述中，用Object1表示此对象的Object\_Identifier。

1. WriteProperty服务正响应执行测试

此测试组的目的是在预期服务成功完成的情况下验证WriteProperty服务请求的正确执行。

1. 写数组中的单个元素

目的：当属性是可写的数组时，验证IUT是否可以执行单个数组元素的WriteProperty服务请求。

测试思想：从被测设备中选择一个可写的数组属性对象，该属性指定为P1。如果找不到符合的对象，则省略该试验。

配置要求：如果被测设备支持写数组值，则应至少配置一个可用于此测试的此类属性。

测试步骤：

1. VERIFY（Object1），P1 = (the value defined for this property in the EPICS)
2. TRANSMIT WriteProperty-Request，

'Object Identifier'= Object1，

'Property Identifier'= P1，

'Property Array Index'= (any value N: 1 ≤ N ≤ the size of the array)

'Property Value'= (any value of the correct datatype for this array except the value verified for this element in step 1)

1. RECEIVE Simple-ACK-PDU
2. VERIFY（Object1），P1 = (the value used in step 2),ARRAY INDEX = N.
3. 写没有优先级的可命令属性

目的：当属性为可命令属性，验证IUT是否可以执行未指定优先级的WriteProperty服务请求。

测试思想：从IUT中选择一个可写的可命令属性对象，并且在优先级为16时没有内部算法写入。如果找不到符合的对象，则应省略该测试。

配置要求：如果被测设备支持优先级为16的没有内部算法写入的可命令属性，则应至少配置一个可用于此测试的此类属性。

测试步骤：

1. VERIFY（Object1），Priority\_Array=(the value defined for this property in the EPICS),ARRAY

INDEX = 16

1. TRANSMIT WriteProperty-Request，

'Object Identifier'= Object1，

'Property Identifier'= Present\_Value，

'Property Value' = (any value of the correct datatype for this property subject to the restrictions specified in the EPICS as defined in 5.4.2, except the value verified in step 1)

1. RECEIVE Simple-ACK-PDU
2. VERIFY (Object1), Priority\_Array =(the value used in step 2), ARRAY INDEX =16
3. 写具有优先级的非命令属性

目的：当属性为非可命令属性，验证IUT是否可以执行指定优先级的 WriteProperty服务请求。

测试思想：从TD中选择一个可写的属性对象，该属性为非可命令属性且没有内部算法修改它。如果找不到符合的属性，则应省略该测试。

配置要求：如果IUT支持没有内部算法写入的不可命令属性，则应至少配置一个可用于此测试的此类属性。

测试步骤：

1. VERIFY(Object1),P1 = (the value defined for this property in the EPICS)
2. TRANSMIT WriteProperty-Request，

'Object Identifier'= Object1

'Property Identifier'= P1，

'Priority' = (any valid priority)

'Property Value' = (any valid value defined for this property subject to the restrictions

specified in the EPICS as defined in 5.4.2, except the value verified in step 1)

1. RECEIVE BACnet-BACnet-SimpleACK-PDU
2. VERIFY (Object1), P1 =(the value used in step 2)
3. WriteProperty服务负响应执行测试

此测试组的目的是在预期服务失败的情况下验证WriteProperty服务请求的正确执行。

1. 用数组索引写非数组属性

目的：当属性值是非数组时，验证IUT是否可以执行包含数组索引的WriteProperty服务请求。

测试思想：从TD中选择一个可写的标量属性，指定为P1，尝试使用数组索引写此属性。如果找不到符合的属性，则应省略该测试。

配置要求：如果IUT支持任何可写的标量属性，则应至少配置一个可用于此测试的此类属性。

测试步骤：

1. VERIFY (Object1), P1 = (the value defined for this property in the EPICS)
2. TRANSMIT WriteProperty-Request,

'Object Identifier' = Object1,

'Property Identifier' = P1,

'Property Value' = (any value of the correct datatype for this property subject to the restrictions specified in the EPICS as defined in 5.4.2, except the value verified in step 1),

'Property Array Index' = (any positive integer)

1. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 4) THEN

RECEIVE BACnet-Error PDU,

Error Class = PROPERTY,

Error Code = PROPERTY\_IS\_NOT\_AN\_ARRAY

ELSE

RECEIVE BACnet-Error PDU,

Error Class = SERVICES,

Error Code = INCONSISTENT\_PARAMETERS

1. VERIFY (Object1), P1 = (the value defined for this property in the EPICS)
2. 使用超出范围的数组索引写数组属性

目的：当属性为数组属性时，验证IUT是否可以执行数组索引超出范围的WriteProperty服务请求。

测试思想：从TD中选择一个可写的数组属性，指定为P1，尝试使用超出数组索引范围写此属性。如果找不到符合的属性，则应省略该测试。

配置要求：如果IUT支持任何可写的数组属性，则应至少配置一个可用于此测试的此类属性。

测试步骤：

1. VERIFY (Object1), P1 = (the value defined for this property in the EPICS)
2. TRANSMIT WriteProperty-Request,

'Object Identifier' = Object1,

'Property Identifier' = P1,

'Property Value' = (any value of the correct datatype for this property subject to the restrictions specified in the EPICS as defined in 5.4.2, except the value verified in step 1),

'Property Array Index' = (any positive integer that is larger that the supported size if the array)

1. RECEIVE BACnet-Error PDU,

Error Class = PROPERTY,

Error Code = INVALID\_ARRAY\_INDEX

1. VERIFY (Object1), P1 = (the value defined for this property in the EPICS)
2. 写入错误数据类型的属性值

目的：验证IUT是否能够正确响应无效数据类型属性值的写入操作。

测试思想：从TD中选择一个可写属性，指定为P1，尝试使用无效的数据类型写此属性。如果没有对象支持可写属性，则应省略该测试。

测试步骤：

1. READ V = (Object1), P1
2. TRANSMIT WriteProperty-Request,

'Object Identifier' = Object1,

'Property Identifier' = P1,

'Property Value'= (any value with an invalid datatype)

1. RECEIVE (BACnet-Error PDU,

Error Class = PROPERTY,

Error Code = INVALID\_DATATYPE)

| (BACnet-Reject-PDU

Reject Reason = INVALID\_PARAMETER\_DATATYPE)

| (BACnet-Reject-PDU

Reject Reason = INVALID\_TAG)

1. VERIFY (Object1), P1 = V
2. 写入超范围的属性值

目的：验证IUT是否可以正确执行超属性值范围的WriteProperty服务请求。

测试思想：尝试使用超出属性范围的值去写该属性。如果IUT不包含任何具有范围限制的可写属性，则应跳过此测试。

测试步骤：

1. VERIFY (Object1), P1 = (the value defined for this property in the EPICS)
2. TRANSMIT WriteProperty-Request,

'Object Identifier' = (Object1, any object with writable properties),

'Property Identifier' = (P1, any writable property with a restricted range of values),

'Property Value' = (any value, of the correct datatype, that is outside the supported range)

1. IF (Protocol\_Revision is present and Protocol\_Revision ≥ 4) THEN

RECEIVE (BACnet-Error PDU,

Error Class = PROPERTY,

Error Code = VALUE\_OUT\_OF\_RANGE

ELSE

RECEIVE (BACnet-Error-PDU,

Error Class = PROPERTY,

Error Code = VALUE\_OUT\_OF\_RANGE)

|(BACnet-Reject-PDU,

Reject Reason = PARAMETER\_OUT\_OF\_RANGE)

1. VERIFY (Object1), P1 = (the value defined for this property in the EPICS)

注意事项：步骤2中使用的值应为正确的数据类型。对于位串类型，日期和时间值的位数应正确，该值应在数据类型标准定义的范围内；对于构造值，构造值应与ASN.1定义的结构相匹配，并且所有字段值应在这些字段值的标准定义的范围内。

1. 写不存在的对象

目的：此测试用例验证当服务请求中指定的对象不存在时，IUT是否可以正确执行WriteProperty服务请求。仅当Protocol\_Revision存在且值大于或等于4时，才应执行此测试。

测试思想：选择一个TD不存在的对象，指定为Object1对象。Object1应为IUT支持的类型。尝试在此不存在的对象中写入指定为P1的属性。P1应为IUT中此对象类型支持的标准属性。

测试步骤：

1. TRANSMIT WriteProperty-Request,

'Object Identifier' = Object1,

'Property Identifier' = P1,

'Property Value'= (any value of the correct datatype for this property)

1. RECEIVE BACnet-Error PDU,

Error Class = OBJECT,

Error Code = UNKNOWN\_OBJECT

传递结果：虽然OBJECT :: UNKNOWN\_OBJECT是此条件的目标的错误，但在某些实现中，可以在对象本身存在之前检查其他错误条件。可接受的其他错误包括：

PROPERTY:: UNKNOWN\_PROPERTY，

PROPERTY:: WRITE\_ACCESS\_DENIED，

PROPERTY:: INVALID\_DATATYPE，

PROPERTY :: VALUE\_OUT\_OF\_RANGE

RESOURCES:: NO\_SPACE\_TO\_WRITE\_PROPERTY.

1. 写入不存在的属性

目的：此测试用例用于验证当服务请求中指定的对象不支持服务请求中指定的属性时，IUT是否可以正确执行 WriteProperty服务请求。 仅当Protocol\_Revision存在且值大于或等于4时，才应执行此测试。

测试思想：从TD中选择一个对象，指定为Object1，选择对象实例中一个不支持的属性，指定为P1，然后尝试写此属性。

测试步骤：

1. TRANSMIT WriteProperty-Request,

'Object Identifier' = Object1,

'Property Identifier' = P1,

'Property Value' = (any valid value for this property)

1. RECEIVE BACnet-Error PDU,

Error Class = PROPERTY,

Error Code = UNKNOWN\_PROPERTY

1. 写不可写属性

目的：此测试用例用于验证当服务请求中指定的属性不可写时，测试设备是否可以正确执行WriteProperty服务请求。仅当Protocol\_Revision存在且值大于或等于4时，才应执行此测试。

测试思想：从TD中选择一个对象，指定为Object1，选择对象实例中一个不支持写的属性，指定为P1，然后尝试写此属性。如果没有对象支持不可写属性，则应省略该测试。

测试步骤：

1. TRANSMIT WriteProperty-Request,

'Object Identifier' = Object1,

'Property Identifier' = P1,

'Property Value' = (any value of the correct datatype for this property)

1. RECEIVE BACnet-Error PDU,

Error Class = PROPERTY,

Error Code = WRITE\_ACCESS\_DENIED

1. 写入已经使用的Object\_Name

目的：此测试用例用于验证当使用设备中其他对象已使用的Object\_Name进行写属性操作时，IUT能否正确执行此WriteProperty服务请求。仅当Protocol\_Revision存在且值大于或等于4时，才应执行此测试。

测试思想：从IUT中选择一个对象，指定为Object1，该对象包含可写的Object\_Name属性。然后尝试使用设备中另一个对象的Object\_Name属性值写此属性。如果没有支持可写Object\_Name属性的对象，则应省略该测试。

测试步骤：

1. TRANSMIT WriteProperty-Request,

'Object Identifier' = Object1,

'Property Identifier' = Object\_Name,

'Property Value' = (an Object\_Name value already in use by another object)

1. RECEIVE BACnet-Error PDU,

Error Class = PROPERTY,

Error Code = DUPLICATE\_NAME

* 1. 写多个属性（WritePropertyMultiple）服务执行测试

本节定义了验证支持执行WritePropertyMultiple服务请求的必需测试项。

所需测试：无。

BACnet参考条款：15.10。

配置要求：WritePropertyMultiple服务执行测试要求IUT具备至少包含两个可写属性的BACnet对象。这些对象的Object\_Identifier在测试描述中被指定为Object1和Object2。

1. WritePropertyMultiple服务正响应执行测试

此测试组的目的是在预期服务成功完成的情况下验证WritePropertyMultiple服务请求的正确执行。

1. 对单个对象写入单个属性

目的：验证是否具备将单个属性写入单个对象的能力。

测试思想：此测试用例尝试对单个不可命令的标量属性P1进行写操作。如果不存在此类可写属性，则可以修改测试写入数组属性或可命令属性，保证使用足够高的写入优先级能够改变可命令属性值。

配置要求：如果IUT支持不可命令的可写标量属性，则应配置一个用于此测试。如果不支持这样的属性，则IUT应配置可写数组或可命令属性，并修改测试步骤以适应此变化。

测试步骤：

1. VERIFY (Object1), P1 = (the value specified for this property in the EPICS)
2. TRANSMIT WritePropertyMultiple-Request,

'Object Identifier' = Object1,

'Property Identifier' = P1,

'Property Value' = (any value of the appropriate datatype except for the one verified in step 1)

1. RECEIVE BACnet-Simple-ACK-PDU
2. VERIFY (Object1), P1 = (the value specified in step 2)
3. 向单个对象写入多个属性值

目的：验证是否具备将多个属性写入单个对象的能力。

测试思想：此测试用例尝试写入多个不可命令的标量属性P1和P2。如果不存在两个这样的可写属性，则可以修改测试用例以写入数组属性或可命令属性，保证使用足够高的写入优先级能够改变可命令属性值。

配置要求：如果IUT支持的对象具有两个不可命令的可写标量属性，则应配置一个这样的对象用于此测试。如果不支持这样的属性，则IUT应尽可能使用可写数组或可命令属性，并修改测试步骤以适应此变化。如果不支持具有两个或更多可写属性的对象类型，则可以省略该测试。IUT必须支持此测试配置或10.23.1.3测试配置。

测试步骤：

1. VERIFY (Object1), P1 = (the value specified for this property in the EPICS)
2. VERIFY (Object1), P2 = (the value specified for this property in the EPICS)
3. TRANSMIT WritePropertyMultiple-Request,

'Object Identifier' = Object1,

'Property Identifier' = P1,

'Property Value' = (any value of the appropriate datatype except for the one verified in step 1),

'Property Identifier' = P2,

'Property Value' = (any value of the appropriate datatype except for the one verified in step 2)

1. RECEIVE BACnet-Simple-ACK-PDU
2. VERIFY (Object1), P1 = (the value specified for P1 in step 2)
3. VERIFY (Object1), P2 = (the value specified for P2 in step 2)
4. 将单个属性写入多个对象

目的：验证是否具备将单个属性写入多个对象的能力。

测试思想：此测试用例尝试将不可命令的单个标量属性P1和P2写入到不同对象中。如果IUT不存在两个这样的可写属性，则可以修改测试用例以写入数组属性或可命令属性，保证使用足够高的写入优先级能够改变可命令属性值。

测试步骤：

1. VERIFY (Object1), P1 = (the value specified for this property in the EPICS)
2. VERIFY (Object2), P2 = (the value specified for this property in the EPICS)
3. TRANSMIT WritePropertyMultiple-Request,

'Object Identifier' = Object1,

'Property Identifier' = P1,

'Property Value' = (any value of the appropriate datatype except forthe one verified in step 1),

'Object Identifier' = Object2,

'Property Identifier' = P2,

'Property Value' = (any value of the appropriate datatype except for the one verified in step 2)

1. RECEIVE BACnet-Simple-ACK-PDU
2. VERIFY (Object1), P1 = (the value specified for P1 in step 3)
3. VERIFY (Object2), P2 = (the value specified for P2 in step 3)
4. 将多个属性写入多个对象

目的：验证是否具备将多个属性写入多个对象的能力。

测试思想：此测试用例尝试写入Object1中的属性P1和P2，以及Object2中的属性P3和P4，其中P1，P2，P3和P4为非可命令的属性。如果不存在四个这样的可写属性，则可以修改测试用例以写入数组属性或可命令属性，保证使用足够高的写入优先级能够改变可命令属性值。

测试步骤：

1. VERIFY (Object1), P1 = (the value specified for this property in the EPICS)
2. VERIFY (Object1), P2 = (the value specified for this property in the EPICS)
3. VERIFY (Object2), P3 = (the value specified for this property in the EPICS)
4. VERIFY (Object2), P4 = (the value specified for this property in the EPICS)
5. TRANSMIT WritePropertyMultiple-Request,

'Object Identifier' = Object1,

'Property Identifier' = P1,

'Property Value' = (any value of the appropriate datatype except for the one verified in step 1),

'Property Identifier' = P2,

'Property Value' = (any value of the appropriate datatype except for the one verified in step 2),

'Object Identifier' = Object2,

'Property Identifier' = P3,

'Property Value' = (any value of the appropriate datatype except for the one verified in step 3),

'Object Identifier' = Object2,

'Property Identifier' = P4,

'Property Value' = (any value of the appropriate datatype except for the one verified in step 4)

1. RECEIVE BACnet-BACnet-SimpleACK-PDU
2. VERIFY (Object1), P1 = (the value specified for P1 in step 5)
3. VERIFY (Object1), P2 = (the value specified for P2 in step 5)
4. VERIFY (Object2), P3 = (the value specified for P3 in step 5)
5. VERIFY (Object2), P4 = (the value specified for P4 in step 5)
6. 写具有优先级的非可命令属性

目的：当属性为非可命令属性时，验证IUT是否可以执行指定优先级的WritePropertyMultiple服务请求。

测试思想：使用指定优先级的WritePropertyMultiple服务而不是WriteProperty服务重复10.22.1.3项测试。

1. 写不指定优先级的可命令属性

目的：当属性为可命令属性时，验证IUT是否可以执行不指定优先级的WritePropertyMultiple服务请求。

测试思想：不指定优先级情况下，使用WritePropertyMultiple服务而不是WriteProperty服务重复10.22.1.2项测试。

1. 写最大多个属性

目的：此测试用例用于验证IUT使用单个WritePropertyMultiple服务请求没有任意的限制属性数量。

测试思想：向IUT的对象中的一个可写属性写入最大可接受请求的长度限制的次数。最大可接受请求大小的计算应基于IUT的Max\_APDU\_Length\_Accepted和每个请求的最大段数。

确定要使用的值的数量的过程是：

MaxAPDU = IUT’s Max\_APDU\_Length\_Accepted

MaxRxSegs = IUT’s maximum segments accepted per request

MaxTxSegs = IUT’s maximum segments generated per response

NonSegRqstHdrSize = size of (non-segmented BACnetConfirmed-RequestPDU header) = 4

SegRqstHdrSize = size of (segmented BACnetConfirmed-RequestPDU header) = 6

ObjIdSize = size of (an Object-Identifier) = 5

TagsSize = size of (an open and a close tag) = 2

PropIdSize = size of (chosen property Id) = depends on property ID and includes array index size if

required

ValueSize = size of (chosen property value) = depends on property and value chosen

如果IUT不支持接收分段请求：

NumPropertiesToWrite=

(MaxAPDU-NonSegRqstHdrSize-ObjIdSize-TagsSize)/(PropIdSize+TagsSize+ValueSize)=

(MaxAPDU - 11)/(PropIdSize + 2 + ValueSize)；

如果IUT支持接收分段请求：

NumPropertiesToWrite=

((MaxAPDU - SegRqstHdrSize)\* MaxRxSegs - ObjIdSize - TagsSize)/ PropIdSiz=

((MaxAPDU - 6)\* MaxRxSegs - 7)/ 2

测试步骤：

1. TRANSMIT WritePropertyMultiple-Request,

'Object Identifier' = (Device, X),

'Property Identifier' = P1,

'Array Index' = A1, -- only if required

'Property Value' = V1,

'Property Identifier' = P1,

'Array Index' = A1, -- only if required

'Property Value' = V1

1. RECEIVE Simple-ACK
2. VERIFY (P1 = V1)
3. 基于数据类型写入属性

目的：此测试用例用于验证IUT是否可以对被设备支持的任何数据类型执行WritePropertyMultiple服务请求。

测试思想：对于指定的数据类型，从TD中选择包含该数据类型的可写属性的对象。该属性指定为P1。

配置要求：IUT配置至少一个指定类型的可写属性用于此测试。

测试步骤：

1. READ V = Object1, P1
2. TRANSMIT WritePropertyMultiple-Request,

'Object Identifier' = Object1,

'Property Identifier' = P1,

'Property Value' = (any valid value defined for this property subject to the restrictions specified in the EPICS as defined in Clause 5.4.2, except the value, V, read in step 1)

1. RECEIVE Simple-ACK-PDU
2. VERIFY (Object1), P1 =(the value used in step 2)
3. WritePropertyMultiple服务负响应执行测试

此测试组的目的是在预期服务失败的情况下验证WritePropertyMultiple服务请求的正确执行。

1. 用属性访问错误写多个属性

目的：当“List of Write Access Specifications”包含不受支持的属性时，验证是否能够正确执行WritePropertyMultiple服务请求。

测试思想：尝试在单个对象中写入两个属性。第一个属性是此对象支持和可写的，第二个属性是此对象不支持的。目标是验证是否在第一次尝试写入失败后返回相应的错误响应。

配置要求：如果IUT支持任何非可命令的可写标量属性，则应配置一个用于此测试。如果不支持这样的属性，则IUT应配置可写数组或可命令属性，并修改测试步骤以适应此变化。在测试描述中，Object1将用于指定的对象，P1用于指定可写属性，P2用于指定用于此测试的不支持属性。

测试步骤：

1. VERIFY (Object1), P1 = (the value specified for this property in the EPICS)
2. TRANSMIT WritePropertyMultiple-Request,

'Object Identifier' = Object1,

'Property Identifier' =P1,

'Property Value' = (any value of the appropriate datatype except for the one verified in step 1),

'Property Identifier' =P2,

'Property Value' = (any value of the appropriate datatype)

1. RECEIVE WritePropertyMultiple-Error,

Error Class = PROPERTY,

Error Code = UNKNOWN\_PROPERTY,

objectIdentifier = Object1,

propertyIdentifier = P2

1. VERIFY (Object1), P1 = (the value specified for P1 in step 2)
2. 用对象访问错误写多个属性

目的：当“List of Write Access Specifications”包含不受支持的对象时，验证是否能够正确执行WritePropertyMultiple服务请求。

测试思想：尝试写入两个不同对象中的单个属性。第一个对象是IUT支持的，且属性是可写的；第二个对象是IUT不支持的。目标是验证是否在第一次尝试写入失败后返回相应的错误响应。

配置要求：如果IUT支持任何非可命令的可写标量属性，则应配置一个用于此测试。如果不支持这样的属性，则IUT应配置可写数组或可命令属性，并修改测试步骤以适应此变化。在测试描述中，Object1和P1将为指定用于此测试的可写对象和属性，名称BadObject用于指示不受支持的对象。

测试步骤：

1. VERIFY (Object1), P1 = (the value specified for this property in the EPICS)
2. TRANSMIT WritePropertyMultiple-Request,

'Object Identifier' = Object1,

'Property Identifier' =P1,

'Property Value' = (any value of the appropriate datatype except for the one verified in step 1),

'Object Identifier' = BadObject,

'Property Identifier' =P2,

'Property Value' = (any value of the appropriate datatype)

1. RECEIVE WritePropertyMultiple-Error,

Error Class = OBJECT,

Error Code = UNKNOWN\_OBJECT,

objectIdentifier = BadObject,

propertyIdentifier = P2

1. VERIFY (Object1), P1 = (the value specified for P1 in step 2)
2. 用写访问错误写多个属性

目的：当“List of Write Access Specifications”包含只读属性时，验证正确执行WritePropertyMultiple服务请求的能力。

测试思想：尝试在单个对象中写入两个属性，第一个是对象支持的可写属性，第二个是对象支持的只读属性。目标是验证是否在第一次尝试写入失败后返回相应的错误响应。

配置要求：如果IUT支持任何非可命令的可写标量属性，则应配置一个用于此测试。如果IUT不支持这样的属性，则应配置可写数组或可命令属性，并修改测试步骤以适应此变化。在测试描述中，Object1为指定对象，P1为指定可写的属性，P2为指定只读属性。

测试步骤：

1. VERIFY (Object1), P1 = (the value specified for this property in the EPICS)
2. VERIFY (Object1), P2 = (the value specified for this property in the EPICS)
3. TRANSMIT WritePropertyMultiple-Request,

'Object Identifier' = Object1,

'Property Identifier' =P1,

'Property Value' = (any value of the appropriate datatype except forthe one verified in step 1),

'Property Identifier' =P2,

'Property Value' = (any value of the appropriate datatype except for the one verified in step 1)

1. RECEIVE WritePropertyMultiple-Error,

Error Class = PROPERTY,

Error Code = WRITE\_ACCESS\_DENIED,

objectIdentifier = Object1,

propertyIdentifier = P2

1. VERIFY (Object1), P1 = (the value specified for P1 in step 3)
2. VERIFY (Object1), P2 = (the value specified for this property in the EPICS)
3. 用数组索引写非数组属性

目的：此测试用例验证当属性值不是数组但服务请求中包含数组索引时，测试设备是否可以执行WritePropertyMultiple服务请求。此测验仅当Protocol\_Revision存在且值大于或等于4时才执行。

测试思想：从TD中选择一个名为Object1的对象，该对象包含一个指定的可写标量属性为P1。尝试使用数组索引写此属性。如果找不到合适的属性，则应省略该试验。

配置要求：如果IUT支持任何可写的标量属性，则应配置至少一个可用于该测试的此类属性。

测试步骤：

1. VERIFY (Object1), P1 = (the value defined for this property in the EPICS)
2. TRANSMIT WritePropertyMultiple-Request,

'Object Identifier' = Object1,

'Property Identifier' = P1,

'Property Value' = (any value of the correct datatype for this property subject to the restrictions specified in the EPICS as defined in 5.4.2, except the value verified in step 1),

'Property Array Index' = (any positive integer)

1. RECEIVE WritePropertyMultiple-Error,

Error Class = PROPERTY,

Error Code = PROPERTY\_IS\_NOT\_AN\_ARRAY,

objectIdentifier = Object1,

propertyIdentifier = P1

1. VERIFY (Object1), P1 = (the value defined for this property in the EPICS)
2. 用超出数组索引范围写数组属性

目的：此测试用例验证当请求的属性值是数组但数组索引超出范围时，IUT是否可以执行WritePropertyMultiple服务请求。此测试仅当Protocol\_Revision存在且值大于或等于4时才应执行。

测试思想：从TD中选择一个名为Object1的对象，该对象包含一个指定的可写数组属性为P1。尝试使用超出范围的数组索引写此属性。如果找不到合适的属性，则应省略该试验。

配置要求：如果IUT支持任何可写的数组属性，则应配置至少一个可用于该测试的此类属性。

测试步骤：

1. VERIFY (Object1), P1 = (the value defined for this property in the EPICS)
2. TRANSMIT WritePropertyMultiple-Request,

'Object Identifier' = Object1,

'Property Identifier' = P1,

'Property Value' = (any value of the correct datatype for this property subject to the restrictions specified in the EPICS as defined in 5.4.2, except the value verified in step 1),

'Property Array Index' = (any positive integer that is larger that the supported size of the array)

1. RECEIVE WritePropertyMultiple-Error,

Error Class = PROPERTY,

Error Code = INVALID\_ARRAY\_INDEX,

objectIdentifier = Object1,

propertyIdentifier = P1

1. VERIFY (Object1), P1 = (the value defined for this property in the EPICS)
2. 用错误数据类型写属性值

目的：此测试用例验证向IUT尝试写入具有无效数据类型的属性值时是否能够正确响应。

测试思想：从TD中选择一个名为Object1的对象，该对象包含一个指定的可写属性为P1。尝试使用IUT支持但对属性无效的数据类型写此属性。如果IUT没有对象支持可写属性，则应省略该测试。

测试步骤：

1. READ V = (Object1), P1
2. TRANSMIT WritePropertyMultiple-Request,

'Object Identifier' = Object1,

'Property Identifier' = P1,

'Property Value' = (any value with an invalid datatype)

1. RECEIVE WritePropertyMultiple-Error,

Error Class = PROPERTY,

Error Code = INVALID\_DATATYPE,

objectIdentifier = Object1,

propertyIdentifier = P1

| ( BACnet-Reject-PDU

'Reject Reason' = INVALID\_PARAMETER\_DATATYPE)

| (BACnet-Reject-PDU

'Reject Reason' = INVALID\_TAG)

1. VERIFY (Object1), P1 = (the value defined for this property in the EPICS)
2. 写超出范围的属性值

目的：此测试用例验证当尝试向IUT写入超出支持范围的属性值时，IUT是否可以执行WritePropertyMultiple服务请求。此测试仅当Protocol\_Revision存在且值大于或等于4时才应被执行。

测试思想：从TD中选择一个名为Object1的对象，该对象包含一个指定的可写属性为P1。尝试使用超出支持范围的值来写被测对象的此属性。

测试步骤：

1. VERIFY (Object1), P1 = (the value defined for this property in the EPICS),
2. TRANSMIT WritePropertyMultiple-Request,

'Object Identifier' = (Object1, any object with writable properties),

'Property Identifier' = (P1, any property with a restricted range of values),

'Property Value' = (any value, of the correct datatype, that is outside the supported range)

1. RECEIVE WritePropertyMultiple-Error,

Error Class = PROPERTY,

Error Code = VALUE\_OUT\_OF\_RANGE,

objectIdentifier = Object1,

propertyIdentifier = P1

1. VERIFY (Object1) , P1 = (the value defined for this property in the EPICS)
2. 写不存在的对象

目的：此测试用例验证当服务请求中指定的对象不存在时，IUT是否可以执行WritePropertyMultiple服务请求。此测试仅当Protocol\_Revision存在且值大于或等于4时才应被执行。

测试思想：从TD选择一个不存在的指定为Object1的对象。Object1应是IUT支持的对象类型。尝试在此不存在的对象中写入指定为P1的属性。P1应引用IUT中此对象类型支持的标准属性。

测试步骤：

1. TRANSMIT WritePropertyMultiple-Request,

'Object Identifier' = Object1,

'Property Identifier' = P1,

'Property Value' = (any value of the correct datatype for this property)

1. RECEIVE WritePropertyMultiple-Error,

Error Class = OBJECT,

Error Code = UNKNOWN\_OBJECT,

objectIdentifier = Object1,

propertyIdentifier = P1

测试结果：虽然OBJECT :: UNKNOWN\_OBJECT是此条件的目标的错误，但在某些实现中，可以在对象本身存在之前检查其它错误条件。可接受的其它错误包括：

PROPERTY:: UNKNOWN\_PROPERTY，

PROPERTY:: WRITE\_ACCESS\_DENIED，

PROPERTY:: INVALID\_DATATYPE，

PROPERTY :: VALUE\_OUT\_OF\_RANGE,

RESOURCES:: NO\_SPACE\_TO\_WRITE\_PROPERTY.

1. 写不存在的属性

目的：此测试用例用于验证当服务请求中指定的对象不支持服务请求中指定的属性时，IUT是否可以执行WritePropertyMultiple服务请求。此测试仅当Protocol\_Revision存在且值大于或等于4时才应被执行。

测试思想：从TD中选择一个对象，指定为Object1。选择对象不支持的指定为P1的属性，尝试写此属性。

测试步骤：

1. TRANSMIT WritePropertyMultiple-Request,

'Object Identifier' = Object1,

'Property Identifier' = P1,

'Property Value' = (any value of the correct datatype for the property),

1. RECEIVE BACnet-Error PDU,

Error Class = PROPERTY,

Error Code = UNKNOWN\_PROPERTY,

objectIdentifier = Object1,

propertyIdentifier = P1

1. 写不可写属性

目的：此测试用例用于验证当服务请求中指定的属性不可写时，IUT是否可以执行WritePropertyMultiple服务请求。此测试仅当Protocol\_Revision存在且值大于或等于4时才应被执行。

测试思想：从TD中选择一个指定的对象为Object1，该对象包含一个指定为P1的不可写属性，尝试写此属性。如果没有对象支持不可写属性，则应省略该测试。

测试步骤：

1. TRANSMIT WritePropertyMultiple-Request,

'Object Identifier' = Object1,

'Property Identifier' = P1,

'Property Value' = (any value of the correct datatype for this property)

1. RECEIVE WritePropertyMultiple-Error,

Error Class = PROPERTY,

Error Code = WRITE\_ACCESS\_DENIED,

objectIdentifier = Object1,

propertyIdentifier = P1

1. 用已经使用的Object\_Name进行写操作

目的：此测试用例用于验证当使用IUT中其他对象已使用的值写入Object\_Name属性时，IUT是否可以执行WritePropertyMultiple服务请求。此测试仅当Protocol\_Revision存在且值大于或等于4时才应被执行。

测试思想：从TD中选择一个对象，指定为Object1，该对象在IUT包含可写的Object\_Name属性。尝试使用设备中另一个对象的Object\_Name属性使用的值写入此属性。如果没有对象支持可写的Object\_Name属性，则应省略该测试。

测试步骤：

1. TRANSMIT WritePropertyMultiple-Request,

'Object Identifier' = Object1,

'Property Identifier' = Object\_Name,

'Property Value' = (an Object\_Name value already in use by another object)

1. RECEIVE WritePropertyMultiple-Error,

Error Class = PROPERTY,

Error Code = DUPLICATE\_NAME

* 1. 设备通信控制（DeviceCommunicationControl）服务执行测试

本节定义了验证支持执行DeviceCommunicationControl服务请求所必需的测试。

所需测试：ReadProperty服务执行测试，10.18。

BACnet参考条款：16.1。

1. DeviceCommunicationControl服务正响应执行测试

此测试组的目的是在预期服务成功完成的情况下验证DeviceCommunicationControl服务请求的正确执行。设X是IUT的Device对象的实例编号。

配置要求：如果IUT要求使用有密码的设备通信控制，则应在本节的测试用例中提供有效密码。如果IUT不提供密码保护，则“Password”参数由测试人员自行决定使用任意密码或省略。注意：如果未提供密码保护，则应忽略密码。

1. 无限时间期限下通过DeviceCommunicationControl恢复通信

目的：当“Time Duration”参数设置为无限，使用DeviceCommunicationControl服务恢复通信时，验证DeviceCommunicationControl服务请求服务过程是否正确执行。

测试步骤：

1. TRANSMIT DeviceCommunicationControl-Request,

'Enable/Disable' = DISABLE,

'Password' = (any appropriate password as described in the Test Concept)

1. RECEIVE BACnet-Simple-ACK-PDU
2. WAIT **Internal Processing Fail Time**
3. TRANSMIT ReadProperty-Request,

'Object Identifier' = (Device, X),

'Property Identifier' = (any required non-array property of the Device object)

1. WAIT (an arbitrary time > **Internal Processing Fail Time** selected by the tester)
2. CHECK (Verify that the IUT has not transmitted any messages since the acknowledgment in step 2)
3. TRANSMIT DeviceCommunicationControl-Request,

'Enable/Disable' = ENABLE,

'Password' = (any appropriate password as described in the

Configuration Requirements)

1. RECEIVE BACnet-Simple-ACK-PDU
2. VERIFY (Device, X), (any required non-array property) = (the value for this property specified in the EPICS)
3. 无限期限时间下通过重新初始化设备(ReinitializeDevice)恢复通信

目的：当“Time Duration”参数设置为无限，使用ReinitializeDevice服务恢复通信时，验证DeviceCommunicationControl服务请求服务过程是否正确执行。

所需测试：ReinitializeDevice服务执行测试，10.27

测试步骤：

1. TRANSMIT DeviceCommunicationControl-Request,

'Enable/Disable' = DISABLE,

'Password' = (any appropriate password as described in the Test Concept)

1. RECEIVE BACnet-Simple-ACK-PDU
2. WAIT **Internal Processing Fail Time**
3. TRANSMIT ReadProperty-Request,

'Object Identifier' = (Device, X),

'Property Identifier' = (any required non-array property of the Device object)

1. WAIT (an arbitrary time > **Internal Processing Fail Time** selected by the tester)
2. CHECK (Verify that the IUT has not transmitted any messages since the acknowledgment in step 2)
3. TRANSMIT ReinitializeDevice-Request,

'Reinitialized State of Device' = WARMSTART,

'Password' = (any appropriate password as described in the Test Concept)

1. RECEIVE BACnet-Simple-ACK-PDU
2. CHECK (Did the IUT perform a WARMSTART reboot?)
3. VERIFY (Device, X),

(any required non-array property) = (the value for this property specified in the EPICS)

1. 有限持续时间

目的：当“Time Duration”参数设置为有限时，验证设DeviceCommunicationControl请求服务过程是否正确执行。

测试步骤：

1. TRANSMIT DeviceCommunicationControl-Request,

'Time Duration' = ( T > 1, in minutes, selected by the tester),

'Enable/Disable' = DISABLE,

'Password' = (any appropriate password as described in the Test Concept)

1. RECEIVE BACnet-Simple-ACK-PDU
2. WAIT **Internal Processing Fail Time**
3. TRANSMIT ReadProperty-Request,

'Object Identifier' = (Device, X),

'Property Identifier' = (any required non-array property of the Device object)

1. WAIT (T )
2. CHECK (Verify that the IUT did not transmit any messages between the acknowledgment in step 2

and expiration of timer T)

1. VERIFY (Device, X),

(any required non-array property) = (the value for this property specified in the EPICS)

1. 有限持续时间下通过DeviceCommunicationControl恢复通信

目的：当“Time Duration”参数设置为有限，使用DeviceCommunicationControl服务恢复通信时，验证DeviceCommunicationControl服务请求服务过程是否正确执行。

测试步骤：

1. TRANSMIT DeviceCommunicationControl-Request,

'Time Duration' = ( T > 1, in minutes, selected by the tester),

'Enable/Disable' = DISABLE,

'Password' = (any appropriate password as described in the Test Concept)

1. RECEIVE BACnet-SimpleACK-PDU
2. WAIT **Internal Processing Fail Time**
3. TRANSMIT ReadProperty-Request,

'Object Identifier' = (Device, X),

'Property Identifier' = (any required non-array property of the Device object)

1. WAIT (an arbitrary time > **Internal Processing Fail Time** selected by the tester, and < T as specified in the DeviceCommunicationControl-Request)
2. CHECK (Verify that the IUT has not transmitted any messages since the acknowledgment in step 2)
3. TRANSMIT DeviceCommunicationControl-Request,

'Enable/Disable' = ENABLE,

'Password' = (any appropriate password as described in the Configuration Requirements)

1. RECEIVE BACnet-SimpleACK-PDU
2. VERIFY ((Device, X), (any required non-array property) = (the value for this property as described in

the Configuration Requirements))

1. 有限持续时间下通过ReinitializeDevice服务恢复通信

目的：当“Time Duration”参数设置为有限，使用ReinitializeDevice服务恢复通信时，验证DeviceCommunicationControl服务请求服务过程是否正确执行。

测试步骤：

1. TRANSMIT DeviceCommunicationControl-Request,

'Time Duration' = (a value T > 1, in minutes, selected by the tester)

'Enable/Disable' = DISABLE,

'Password' = (any appropriate password as described in the Test Concept)

1. RECEIVE BACnet-SimpleACK-PDU
2. WAIT **Internal Processing Fail Time**
3. TRANSMIT ReadProperty-Request,

'Object Identifier' = (Device, X),

'Property Identifier' = (any required non-array property of the Device object)

1. WAIT (an arbitrary time > **Internal Processing Fail Time** selected by the tester, and < T as specified

in the DeviceCommunicationControl-Request)

1. CHECK (Verify that the IUT has not transmitted any messages since the acknowledgment in step 2.)
2. TRANSMIT ReinitializeDevice-Request,

'Reinitialize State of Device' = WARMSTART,

'Password' = (any appropriate password as described in the Test Concept)

1. RECEIVE BACnet-Simple-ACK-PDU
2. CHECK (Did the IUT perform a WARMSTART reboot?)
3. VERIFY (Device, X), (any required non-array property) = (the value for this property as described in

the EPICS)

1. 无限期持续时间，禁用启动，通过DeviceCommunicationControl恢复通信

目的：此测试用例用于当“Time Duration”参数设置为无限，“Enable/Disable”参数设置为DISABLE\_INITIATION时，使用DeviceCommunicationControl服务恢复通信，验证DeviceCommunicationControl服务请求服务过程是否正确执行。如果IUT在响应Who-Is时没有启动I-Am以外的任何服务，则应跳过此测试。

测试步骤：

1. TRANSMIT DeviceCommunicationControl-Request,

'Enable/Disable' = DISABLE-INITIATION,

'Password' = (any appropriate password as described in the Test Concept)

1. RECEIVE BACnet-Simple-ACK-PDU
2. WAIT **Internal Processing Fail Time**
3. MAKE (do something that would normally cause the IUT to initiate a message)
4. WAIT (an arbitrary time > **Internal Processing Fail Time** selected by the tester)
5. CHECK (Verify that the IUT has not transmitted any messages since the acknowledgment in step 2.)
6. TRANSMIT ReadProperty-Request,

'Object Identifier' = (Device, X),

'Property Identifier' = (any required non-array property of the Device object)

1. RECEIVE BACnet-ComplexACK-PDU,

'Object Identifier' = (Device, X),

'Property Identifier' = (the 'Property Identifier' specified in step 7),

'Property Value' = (any valid value for the property)

1. TRANSMIT DeviceCommunicationControl-Request,

'Enable/Disable' = ENABLE,

'Password' = (any appropriate password as described in the Test Concept)

1. RECEIVE BACnet-Simple-ACK-PDU
2. MAKE (do something to cause the IUT to initiate a message)
3. WAIT **Internal Processing Fail Time**
4. CHECK (Verify that the IUT initiated a message)
5. 无限期持续时间，禁用启动，通过ReinitializeDevice服务恢复通信

所需测试：重新初始化设备服务执行测试，10.27。

目的：此测试用例用于当“Time Duration”参数设置为无限，“Enable/Disable”参数设置为DISABLE\_INITIATION时，使用ReinitializeDevice服务恢复通信，验证DeviceCommunicationControl服务请求服务过程是否正确执行。如果IUT在响应Who-Is时没有启动I-Am以外的任何服务，则应跳过此测试。

测试步骤：

1. TRANSMIT DeviceCommunicationControl-Request,

'Enable/Disable' = DISABLE-INITIATION,

'Password' = (any appropriate password as described in the Test Concept)

1. RECEIVE BACnet-Simple-ACK-PDU
2. WAIT **Internal Processing Fail Time**
3. MAKE (do something that would normally cause the IUT to initiate a message)
4. WAIT (an arbitrary time > **Internal Processing Fail Time** selected by the tester)
5. CHECK (Verify that the IUT has not transmitted any messages since the acknowledgment in step 2.)
6. TRANSMIT ReadProperty-Request,

'Object Identifier' = (Device, X),

'Property Identifier' = (any required non-array property of the Device object)

1. RECEIVE BACnet-ComplexACK-PDU,

'Object Identifier' = (Device, X),

'Property Identifier' = (the 'Property Identifier' specified in step 7),

'Property Value' = (any valid value for the property)

1. TRANSMIT ReinitializeDevice-Request,

'Reinitialized State of Device' = WARMSTART,

'Password' = (any appropriate password as described in the Test Concept)

1. RECEIVE BACnet-Simple-ACK-PDU
2. CHECK (Did the IUT perform a WARMSTART reboot?)
3. MAKE (do something to cause the IUT to initiate a message)
4. WAIT **Internal Processing Fail Time**
5. CHECK (Verify that the IUT initiated a message)
6. 有限持续时间，禁用启动

目的：此测试用例用于当“Time Duration”参数设置为有限，“Enable/Disable”参数设置为DISABLE\_INITIATION时，验证DeviceCommunicationControl服务请求服务过程是否正确执行。如果IUT在响应Who-Is时没有启动I-Am以外的任何服务，则应跳过此测试。

测试步骤：

1. TRANSMIT DeviceCommunicationControl-Request,

'Time Duration' = (a value T > 1, in minutes, selected by the tester),

'Enable/Disable' = DISABLE-INITIATION,

'Password' = (any appropriate password as described in the Test Concept)

1. RECEIVE BACnet-Simple-ACK-PDU
2. WAIT **Internal Processing Fail Time**
3. MAKE (do something that would normally cause the IUT to initiate a message)
4. WAIT (an arbitrary time > **Internal Processing Fail Time** selected by the tester)
5. CHECK (Verify that the IUT has not transmitted any messages since the acknowledgment in step 2.)
6. TRANSMIT ReadProperty-Request,

'Object Identifier' = (Device, X),

'Property Identifier' = (any required non-array property of the Device object)

1. RECEIVE BACnet-ComplexACK-PDU,

'Object Identifier' = (Device, X),

'Property Identifier' = (the 'Property Identifier' specified in step 7),

'Property Value' = (any valid value for the property)

1. WAIT (T)
2. MAKE (do something to cause the IUT to initiate a message)
3. WAIT **Internal Processing Fail Time**
4. CHECK (Verify that the IUT initiated a message)
5. 禁用启动，通过有限持续时间到达恢复

BACnet参考条款：16.1.1.1.2。

目的：当“Time Duration”参数设置为有限，“Enable/Disable”参数设置为DISABLE\_INITIATION时，验证DeviceCommunicationControl服务中当“Time Duration”参数到期时能否正确恢复通信。

配置要求：IUT应配置为发起请求客户端。

测试步骤：

1. MAKE (a condition that would normally cause the IUT to initiate requests)
2. CHECK (that the IUT is initiating requests)
3. TRANSMIT DeviceCommunicationControl-Request,

'Time Duration' = (a value T > 1, in minutes, selected by the tester),

'Enable/Disable' = DISABLE\_INITIATION,

'Password' = (any appropriate password if required)

1. RECEIVE BACnet-SimpleACK-PDU
2. MAKE (a condition that would normally cause the IUT to initiate requests)
3. CHECK (that the IUT has stopped initiating requests)
4. VERIFY (any supported property) = (any valid value)
5. TRANSMIT Who-Is-Request
6. RECEIVE I-Am-Request
7. WAIT (time T)
8. MAKE (a condition that would normally cause the IUT to initiate requests)
9. CHECK (that the IUT is initiating requests)

注意事项：步骤2到9必须在时间T内执行。

1. 禁用启动，通过DeviceCommunicationControl服务恢复

BACnet参考条款：16.1.1.1.2。

目的：当“Time Duration”参数设置为有限，“Enable/Disable”参数设置为DISABLE\_INITIATION时，验证DeviceCommunicationControl服务能否正确恢复通信。

配置要求：IUT应配置为发起请求客户端。

测试步骤：

1. MAKE (a condition that would normally cause the IUT to initiate requests)
2. CHECK (that the IUT is initiating requests)
3. TRANSMIT DeviceCommunicationControl-Request,

'Time Duration' = (a value in minutes > time required to execute all test steps),

'Enable/Disable' = DISABLE\_INITIATION,

'Password' = (any appropriate password if required)

1. RECEIVE BACnet-SimpleACK-PDU
2. MAKE (a condition that would normally cause IUT to initiate requests)
3. CHECK (that the IUT has stopped initiating requests)
4. VERIFY (any supported property) = (any valid value)
5. TRANSMIT Who-Is-Request
6. RECEIVE I-Am-Request
7. TRANSMIT DeviceCommunicationControl-Request,

'Enable/Disable' = ENABLE,

'Password' = (any appropriate password if required)

1. MAKE (a condition that would normally cause the IUT to initiate requests)
2. CHECK (that the IUT is initiating requests)
3. 负响应DeviceCommunicationControl服务执行测试

此测试组的目的是在预期服务失败的情况下验证DeviceCommunicationControl服务请求的正确执行。

1. 无效密码

目的：验证在提供无效密码时DeviceCommunicationControl服务过程能否正确执行。如果IUT不提供密码保护，则应省略该测试。

测试步骤：

1. TRANSMIT DeviceCommunicationControl-Request,

'Enable/Disable' = DISABLE,

'Password' = (any invalid password)

1. RECEIVE BACnet-Error-PDU,

Error Class = SECURITY,

Error Code = PASSWORD\_FAILURE

1. VERIFY (Device, X),

(any required non-array property) = (the value for this property specified in the EPICS)

1. 缺少密码

目的：在密码保护的情况下，验证使用缺失密码的DeviceCommunicationControl服务过程是否能正确执行。如果IUT不提供密码保护，则应省略该测试。

测试步骤：

1. TRANSMIT DeviceCommunicationControl-Request,

'Enable/Disable' = DISABLE,

1. IF (Protocol\_Revision >= 7) THEN

RECEIVE BACnet-Error-PDU,

Error Class = SECURITY,

Error Code = PASSWORD\_FAILURE

ELSE

RECEIVE BACnet-Error-PDU,

Error Class = SECURITY,

Error Code = PASSWORD\_FAILURE

|(RECEIVE BACnet-Error-PDU,

Error Class = SERVICES,

Error Code = MISSING\_REQUIRED\_PARAMETER)

1. VERIFY (Device,X), System\_Status = (any valid value)
2. 使用无效的“Reinitialized State of Device”参数进行ReinitializeDevice恢复通信

目的：当“Reinitialized State of Device”参数为备份或恢复相关值时，验证ReinitializeDevice服务不能恢复设备通信。

测试思想：禁用IUT的通信一段时间T，该时间应该比完成测试所需的时间长。验证在通信被禁用时，IUT在收到包含备份或恢复相关值的重新初始化设备请求时能够正确给出负响应。

测试步骤：

1. TRANSMIT DeviceCommunicationControl-Request,

'Enable/Disable' = DISABLE

'Password' = (any appropriate password),

'Time Duration' = (a value T >= 1, in minutes) | (no value)

1. RECEIVE BACnet-Simple-ACK-PDU
2. WAIT **Internal Processing Fail Time**
3. TRANSMIT ReinitializeDevice-Request,

'Reinitialized State of Device' = STARTBACKUP | ENDBACKUP | STARTRESTORE | ENDRESTORE | ABORTRESTORE,

'Password' = (any appropriate password)

1. IF (Protocol\_Revision is present and Protocol\_Revision >= 7) THEN

RECEIVE BACnet-Error-PDU,

Error Class = SERVICES,

Error Code = COMMUNICATION\_DISABLED

ELSE

CHECK(that the IUT responded with BACnet-Error-PDU with an Error Class of SERVICES and an Error Code of COMMUNICATION\_DISABLE, or the IUT did not respond at all)

1. TRANSMIT DeviceCommunicationControl-Request,

'Enable/Disable' = ENABLE

'Password' = (any appropriate password)

1. RECEIVE BACnet-Simple-ACK-PDU
   1. 确认专有转换（ConfirmedPrivateTransfer）服务执行测试

本节定义了验证支持执行ConfirmedPrivateTransfer请求所必需的测试。

所需测试：无。

BACnet参考条款：16.2。

目的：验证正确执行ConfirmedPrivateTransfer服务请求的能力。

测试思想：特定专用传输服务的服务过程由供应商定义。此测试仅简单的验证是否返回了正确的响应，并且遵守供应商定义的任何外部可见操作。

配置要求：IUT应配置为执行至少一个ConfirmedPrivateTransfer服务。还应提供请求的服务参数以及应该对测试人员显而易见的任何外部可见操作的列表。

测试步骤：

1. TRANSMIT ConfirmedPrivateTransfer-Request,

'Vendor ID'= (the Vendor\_Identifier specified in the Device object of the EPICS),

'Service Number' = (any service number provided by the vendor),

'Service Parameters' = (the service parameters provided for this service)

1. RECEIVE ConfirmedPrivateTransfer-ACK,

'Vendor ID' = (the Vendor\_Identifier specified in the Device object of the EPICS),

'Service Number' = (the service number used in step 1),

'Result Block' = (the expected results provided by the vendor)

1. CHECK (Did the externally visible actions take place?)
   1. 未确认专有转换（UnconfirmedPrivateTransfer）服务执行测试

BACnet没有定义用于执行UnconfirmedPrivateTransfer服务的服务过程，因此不需要测试。

* 1. 重新初始化设备(ReinitializeDevice)服务执行测试

本节定义了验证支持执行ReinitializeDevice服务请求所必需的测试。

所需测试：无。

BACnet参考条款：16.4。

1. 正响应ReinitializeDevice服务执行测试

此测试组的目的是在预期服务成功完成的情况下验证ReinitializeDevice服务请求的正确执行。

1. 不需要密码的冷启动(COLDSTART)

目的：验证不需密码的ReinitializeDevice请求服务实现冷启动的过程能否正确执行。

测试步骤：

1. TRANSMIT ReinitializeDevice-Request,

'Reinitialized State of Device' = COLDSTART

2. RECEIVE BACnet-Simple-ACK-PDU

3. CHECK (Did the IUT perform a COLDSTART reboot?)

注意事项：IUT应尽可能使用外部指示（如LED或启动消息流量）确认重新初始化成功。

1. 使用正确密码的冷启动COLDSTART

目的：验证使用正确的密码通过 ReinitializeDevice服务请求实现冷启动的过程是否正确执行。

测试思想：无论IUT是否需要密码保护，都会提供密码。如果IUT提供密码保护，则“Password”参数应包含供应商提供的正确密码。如果IUT不提供密码保护，则“Password”参数应包含任意密码。请注意，如果未提供密码保护，则应忽略密码。见BACnet 16.4.1.1.2。

测试步骤：

1. TRANSMIT ReinitializeDevice-Request,

'Reinitialized State of Device' = COLDSTART,

'Password' = (any appropriate password as described in the Test Concept)

1. RECEIVE BACnet-Simple-ACK-PDU
2. CHECK (Did the IUT perform a COLDSTART reboot?)

测试人员注意事项：IUT应尽可能使用外部指示（如LED或启动消息流量）确认重新初始化成功。

1. 不需要密码的热启动(WARMSTART)

目的：验证不需密码的ReinitializeDevice请求服务实现热启动的过程能否正确执行

测试步骤：

1. TRANSMIT ReinitializeDevice-Request,

'Reinitialized State of Device' = WARMSTART

2. RECEIVE BACnet-Simple-ACK-PDU

3. CHECK (Did the IUT perform a WARMSTART reboot?)

测试人员注意事项：IUT应尽可能使用外部指示（如LED或启动消息流量）确认重新初始化成功。

1. 使用正确密码的热启动

目的：验证使用正确的密码通过 ReinitializeDevice服务请求实现热启动的过程是否正确执行。

测试思想：无论IUT是否需要密码保护，都会提供密码。如果IUT提供密码保护，则“Password”参数应包含供应商提供的正确密码。如果IUT不提供密码保护，则“Password”参数应包含任意密码。请注意，如果未提供密码保护，则应忽略密码。 见BACnet 16.4.1.1.2。

测试步骤：

1. TRANSMIT ReinitializeDevice-Request,

'Reinitialized State of Device' = WARMSTART,

'Password' = (any appropriate password as described in the Test Concept)

1. RECEIVE BACnet-Simple-ACK-PDU
2. CHECK (Did the IUT perform a WARMSTART reboot?)

测试人员注意事项：IUT应尽可能使用外部指示（如LED或启动消息流量）确认重新初始化成功。

1. 负响应ReinitializeDevice服务执行测试

此测试组的目的是在预期服务失败的情况下验证ReinitializeDevice服务请求的正确执行。

1. 无效密码的冷启动

目的：验证使用无效密码进行ReinitializeDevice请求服务实现冷启动的过程是否正确执行。如果IUT不提供密码保护，则应省略该测试。

测试步骤：

1. TRANSMIT ReinitializeDevice-Request,

'Reinitialized State of Device' = COLDSTART,

'Password' = (any invalid password)

1. RECEIVE BACnet-Error-PDU,

Error Class = SECURITY,

Error Code = PASSWORD\_FAILURE

1. CHECK (Did the IUT reboot?)

测试人员注意事项：IUT不得重新初始化。

1. 无效密码的热启动

目的：验证使用无效密码进行ReinitializeDevice请求服务实现热启动的过程是否正确执行。如果IUT不提供密码保护，则应省略该测试。

测试步骤：

1. TRANSMIT ReinitializeDevice-Request,

'Reinitialized State of Device' = WARMSTART,

'Password' = (any invalid password)

1. RECEIVE BACnet-Error-PDU,

Error Class = SECURITY,

Error Code = PASSWORD\_FAILURE

1. CHECK (Did the IUT reboot?)

测试人员注意事项：IUT不得重新初始化。

1. 密码丢失的冷启动

目的：当对有密码保护的设备进行不带密码的冷启动操作时，验证IUT能否返回正确的BACnet Error PDU。

测试步骤：

1. TRANSMIT ReinitializeDevice-Request,

'Reinitialized State of Device' = COLDSTART,

1. IF (Protocol\_Revision >= 7) THEN

RECEIVE BACnet-Error-PDU,

Error Class = SECURITY,

Error Code = PASSWORD\_FAILURE

ELSE

(RECEIVE BACnet-Error-PDU,

Error Class = SECURITY,

Error Code = PASSWORD\_FAILURE) |

(BACnet-Error-PDU,

Error Class = SERIVCES,

Error Code = MISSING\_REQUIRED\_PARAMETER)

1. CHECK (The IUT did NOT perform a COLDSTART reboot)

测试人员注意事项：IUT应尽可能使用外部指示（如LED或启动消息流量）确认重新初始化成功。

1. 10.27.2.X4密码丢失的热启动

目的：当对有密码保护的设备进行不带密码的热启动操作时，验证IUT能否返回正确的BACnet Error PDU。

测试步骤：

1. TRANSMIT ReinitializeDevice-Request,

'Reinitialized State of Device' = WARMSTART,

1. IF (Protocol\_Revision >= 7) THEN

RECEIVE BACnet-Error-PDU,

Error Class = SECURITY,

Error Code = PASSWORD\_FAILURE

ELSE

RECEIVE BACnet-Error-PDU,

Error Class = SECURITY,

Error Code = PASSWORD\_FAILURE |

(BACnet-Error-PDU,

Error Class = SERVICES,

Error Code = MISSING\_REQUIRED\_PARAMETER)

1. CHECK (The IUT did NOT perform a WARMSTART reboot)

测试人员注意事项：IUT应尽可能使用外部指示（如LED或启动消息流量）确认重新初始化成功。

* 1. 确认文本信息（ConfirmedTextMessage）服务执行测试

此测试组的目的是验证ConfirmedTextMessage服务请求的正确执行。收到ConfirmedTextMessage服务请求时，BACnet除了要返回确认外没有定义会发生什么。可能会发生一些供应商特定的外部可见的动作。这些测试验证是否返回了正确的确认以及供应商定义的任何其他操作。

BACnet参考条款：16.5。

1. 没有报文类(Message Class)参数的文本报文

目的：验证在未提供“Message Class”时的ConfirmedTextMessage服务请求能否正确执行。

测试步骤：

1. TRANSMIT ConfirmedTextMessage-Request,

'Text Message Source Device' = TD,

'Message Priority' = NORMAL,

'Message' = (any CharacterString)

1. RECEIVE BACnet-SimpleACK-PDU
2. CHECK (Did any vendor specified action for these circumstances occur?)

测试人员注意事项：IUT应响应指示的消息并执行供应商指定的操作。

1. 带有无符号报文类的文本报文

目的：在“Message Class”参数为Unsigned形式时，验证ConfirmedTextMessage服务请求能否正确执行。

配置要求：供应商应提供受支持的Unsigned报文类列表。

测试步骤：

1. TRANSMIT ConfirmedTextMessage-Request,

'Text Message Source Device' = TD,

'Message Class' = (any Unsigned value from the list provided by the vendor),

'Message Priority' = NORMAL,

'Message' = (any CharacterString)

1. RECEIVE BACnet-SimpleACK-PDU
2. CHECK (Did any vendor specified action for these circumstances occur?)

测试人员注意事项：IUT应响应指示的消息并执行供应商指定的操作。

1. 带有字符串报文类的文本报文

目的：在“Message Class”参数为CharacterString形式时，验证ConfirmedTextMessage服务请求能否正确执行。

配置要求：供应商应提供受支持的CharacterString报文类列表。

测试步骤：

1. TRANSMIT ConfirmedTextMessage-Request,

'Text Message Source Device' = TD,

'Message Class' = (any CharacterString value from the list provided by the vendor),

'Message Priority' = NORMAL,

'Message' = (any CharacterString)

1. RECEIVE BACnet-SimpleACK-PDU
2. CHECK(Did any vendor specified action for these circumstances occur?)

测试人员注意事项：IUT应响应指示的消息并执行供应商指定的操作。

1. 紧急优先级的文本报文

目的：在“Message Priority”参数为紧急优先级时，验证ConfirmedTextMessage服务请求能否正确执行。

测试步骤：

1. TRANSMIT ConfirmedTextMessage-Request,

'Text Message Source Device' = TD,

'Message Class' = (any message class from the lists provided by the vendor),

'Message Priority' = URGENT,

'Message' = (any CharacterString)

1. RECEIVE BACnet-SimpleACK-PDU
2. CHECK(Did any vendor specified action for these circumstances occur?)

测试人员注意事项：IUT应响应指示的消息并执行供应商指定的操作。

* 1. 非确认文本信息(UnconfirmedTextMessage)服务执行测试

BACnet没有定义用于执行UnconfirmedTextMessage服务的服务过程，因此不需要进行任何测试。

* 1. 时间同步（TimeSynchronization）服务执行测试

所需测试：执行ReadProperty服务测试，10.18。

BACnet参考条款：16.7。

1. 正响应TimeSynchronization服务执行测试

此测试组的目的是在预期服务成功完成的情况下验证TimeSynchronization服务请求能否正确执行。

1. 本地广播

目的：验证IUT是否重置其本地时间和日期以响应本地广播TimeSynchronization服务请求。

测试步骤：

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Local\_Date

1. RECEIVE ReadProperty-ACK,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Local\_Date,

'Property Value' = (any valid date referred to as "InitialDate" below)

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Local\_Time

1. RECEIVE ReadProperty-ACK,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Local\_Time,

'Property Value' = (any valid time referred to as "InitialTime" below)

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = UTC\_Offset

1. RECEIVE ReadProperty-ACK,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = UTC\_Offset,

'Property Value' = (any valid offset referred to as "InitialUTC\_Offset" below)

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Daylight\_Savings\_Status

1. RECEIVE ReadProperty-ACK,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Daylight\_Savings\_Status,

'Property Value' = (any valid status referred to as "InitialDaylight\_Savings\_Status" below)

1. TRANSMIT

DA = LOCAL BROADCAST,

SA = TD,

BACnet-Unconfirmed-Request-PDU,

'Service Choice' = TimeSynchronization-Request,

date = (any date other than InitialDate),

time = (any time that does not correspond to InitialTime)

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Local\_Date

1. RECEIVE ReadProperty-ACK,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Local\_Date,

'Property Value' = (the date specified in step 9)

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Local\_Time

1. RECEIVE ReadProperty-ACK,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Local\_Time,

'Property Value' = (the time specified in step 9)

测试人员注意事项：IUT在步骤13中返回的时间值应与步骤9中指定的时间一致（符合EPICS中规定的时间分辨率）。如果IUT返回的时间表明，自收到时间同步请求后，已经过了一小段时间(< 1秒)，则测试结果为通过。如果时间未指定星期几但所有其他字段都正确，则结果应视为通过。

1. 发送给IUT

目的：验证IUT是否响应指向IUT的MAC地址的TimeSynchronization服务请求来重置其本地时间和日期。

测试步骤：除了步骤9中的TimeSynchronization-Request使用IUT的MAC地址作为目的地地址之外，该测试与10.30.1.1相同。

测试人员注意事项：通过的判定结果与10.30.1.1相同。

* 1. UTC时间同步（UTCTimeSynchronization）服务执行测试

所需测试：执行ReadProperty服务测试，10.18。

BACnet参考条款：16.8。

1. 正响应UTCTimeSynchronization服务执行测试

此测试组的目的是预计服务成功完成的情况验证UTCTimeSynchronization服务请求能否正确执行。

1. 本地广播

目的：验证IUT是否重置其本地时间和日期以响应本地广播UTCTimeSynchronization服务请求。

测试步骤：测试步骤与10.30.1.1中的步骤相同，只是在步骤9中使用UTCTimeSynchronization服务请求，并且传送的日期和时间用UTC表示。

测试结果：判定测试通过的结果与10.30.1.1相同，除了步骤9中的日期应与InitialUTC\_Offset相对应，步骤13中的时间应与Initial\_UTC\_Offset和Daylight\_Savings\_Status相对应(如BACnet16.7.2中所定义)。

1. 发送给IUT

目的：当UTCTimeSynchronization服务请求的MAC地址为被测试设备时，验证IUT是否正确重置其本地时间和日期。

测试步骤：此测试与10.30.1.1相同，不同之处在于在步骤9中使用UTCTimeSynchronization请求，并且传送的日期和时间用UTC表示，并且应使用IUT的MAC地址作为目的地来传输UTCTimeSynchronization请求。

注意事项：测试通过的判定依据与10.30.1.1相同。

* 1. Who-Has服务执行测试

此测试组的目的是验证Who-Has服务请求是否正确执行。

所需测试：无。

BACnet参考条款：16.9。

1. 执行本地网络发起的Who-Has服务请求

此测试组的目的是验证从本地网络发起的Who-Has请求服务过程是否能够正确执行。

1. 没有设备范围限定的Object ID版本

目的：在不限制设备范围，指定对象标识符参数情况下，验证IUT是否可以正确响本地广播Who-Has服务请求。

测试步骤：

1. TRANSMIT

DA = LOCAL BROADCAST,

SA = TD,

Who-Has-Request,

'Object Identifier' = (any object identifier specified in the EPICS)

1. WAIT **Internal Processing Fail Time**
2. RECEIVE

DA = LOCAL BROADCAST | GLOBAL BROADCAST,

SA = IUT,

I-Have-Request,

'Device Identifier' = (the IUT's Device object),

'Object Identifier' = (the object identifier specified in step 1),

'Object Name' = (the object name specified in the EPICS for this object)

1. 没有设备范围限制的Object Name版本

目的：验证IUT可以正确响应使用对象名称形式且不限制设备范围的本地广播Who-Has服务请求。

测试步骤：

1. TRANSMIT

DA = LOCAL BROADCAST,

SA = TD,

Who-Has-Request,

'Object Name' = (any object name specified in the EPICS)

1. WAIT **Internal Processing Fail Time**
2. RECEIVE

DA = LOCAL BROADCAST | GLOBAL BROADCAST,

SA = IUT,

I-Have-Request,

'Device Identifier' = (the IUT's Device object),

'Object Identifier' = (the object identifier specified in the EPICS for this object),

'Object Name' = (the object name specified in step 1)

1. IUT的Object ID在设备对象实例范围内

目的：验证IUT可以正确响应本地广播的利用了对象标识符形式并指定了包含IUT的设备范围限制的Who-Has服务请求。

测试步骤：

1. TRANSMIT

DA = LOCAL BROADCAST,

SA = TD,

Who-Has-Request,

'Device Instance Low Limit'= (any value L: 0 ≤ L < the Device object instance

number of the IUT),

'Device Instance High Limit' = (any value H,: H > the Device object instance

number of the IUT),

'Object Identifier' = (any object identifier specified in the EPICS),

1. WAIT **Internal Processing Fail Time**
2. RECEIVE

DA = LOCAL BROADCAST | GLOBAL BROADCAST,

SA = IUT,

I-Have-Request,

'Device Identifier' = (the IUT's Device object),

'Object Identifier' = (the object identifier specified in step 1),

'Object Name' = (the object name specified in the EPICS for this object)

1. IUT的Object ID超出设备对象实例范围

目的：验证IUT是否忽略利用对象标识符形式并指定了不包括IUT的设备范围限制的本地广播Who-Has服务请求。

测试步骤：

1. TRANSMIT

DA = LOCAL BROADCAST,

SA = TD,

Who-Has-Request,

'Device Instance Low Limit' = (any value > 0: the Device object instance number does not fall

in the range between Device Instance Low Limit and Device

Instance High Limit),

'Device Instance High Limit' = (any value > Device Instance Low Limit: the Device object

instance number does not fall in the range between Device

Instance Low Limit and Device Instance High Limit),

'Object Identifier' = (any object identifier specified in the EPICS)

1. WAIT **Internal Processing Fail Time**
2. CHECK (verify that the IUT does not respond)
3. 使用Object Name，被测对象在设备对象实例号范围内

目的：当被测对象设备实例号在参数范围内且Object Name存在的情况下，验证IUT是否可以正确响应本地广播Who-Has服务请求。

测试步骤：

1. TRANSMIT

DA = LOCAL BROADCAST,

SA = TD,

Who-Has-Request,

'Device Instance Low Limit' = (any value L: 0 ≤ L < the Device object instance

number of the IUT),

'Device Instance High Limit' = (any value H: H > the Device object instance

number of the IUT),

'Object Name' = (any object name specified in the EPICS)

1. WAIT **Internal Processing Fail Time**
2. RECEIVE

DA = LOCAL BROADCAST | GLOBAL BROADCAST,

SA = IUT,

I-Have-Request,

'Device Identifier' = (the IUT's Device object),

'Object Identifier' = (the object identifier specified in the EPICS for this object),

'Object Name' = (the object name specified in step 1)

1. 使用Object Name，被测对象超过设备对象实例号范围

目的：当被测对象设备实例号不在参数范围内且Object Name存在的情况下，验证IUT是否可以忽略本地广播Who-Has服务请求。

测试步骤：

1. TRANSMIT

DA = LOCAL BROADCAST,

SA = TD,

Who-Has-Request,

'Device Instance Low Limit' = (any value > 0 such that the Device object instance

number does not fall in the range between Device

Instance Low Limit and Device Instance High Limit),

'Device Instance High Limit' = (any value > Device Instance Low Limit such that the Device

object instance number does not fall inthe range between Device

Instance Low Limit and Device Instance High Limit),

'Object Name = (any object name specified in the EPICS)

1. WAIT **Internal Processing Fail Time**
2. CHECK (verify that the IUT does not respond)
3. 使用Object ID，被测对象的设备实例号等于参数范围的上限

目的：当IUT的对象实例号等于参数上限且Object ID存在的情况下，验证IUT能否正确识别Who-Has服务请求。

测试步骤：

1. TRANSMIT

DA = LOCAL BROADCAST,

SA = TD,

Who-Has-Request,

'Device Instance Low Limit' = (any value L: 0 ≤ L < the Device object instance

number of the IUT),

'Device Instance High Limit' = (The Device object instance number of the IUT),

'Object Identifier' = (any object identifier specified in the EPICS)

1. WAIT **Internal Processing Fail Time**
2. RECEIVE

DA = LOCAL BROADCAST | GLOBAL BROADCAST,

SA =IUT,

I-Have-Request,

'Device Identifier' = (the IUT's Device object),

'Object Identifier' = (the object identifier specified in step 1),

'Object Name' = (the object name specified in the EPICS for this object)

1. 使用Object ID，被测对象的设备实例号等于参数范围的下限

目的：当IUT的对象实例号等于参数下限且Object ID存在的情况下，验证IUT能否正确识别Who-Has服务请求。

测试步骤：

1. TRANSMIT

DA = LOCAL BROADCAST,

SA = TD,

Who-Has-Request,

'Device Instance Low Limit' = (The Device object instance number of the IUT),

'Device Instance High Limit' = (any value H: H > the Device object instance

number of the IUT),

'Object Identifier' = (any object identifier specified in the EPICS)

1. WAIT **Internal Processing Fail Time**
2. RECEIVE

DA = LOCAL BROADCAST | GLOBAL BROADCAST,

SA = IUT,

I-Have-Request,

'Device Identifier' = (the IUT's Device object),

'Object Identifier' = (the object identifier specified in step 1),

'Object Name' = (the object name specified in the EPICS for this object)

1. 使用Object Name，被测对象的设备实例号等于参数范围的上限

目的：当IUT的对象实例号等于参数上限且Object Name存在的情况下，验证IUT能否正确识别Who-Has服务请求。

测试步骤：

1. TRANSMIT

DA = LOCAL BROADCAST,

SA = TD,

Who-Has-Request,

'Device Instance Low Limit'= (any value L: 0 ≤ L < the Device object instance

number of the IUT),

'Device Instance High Limit' = (The Device object instance number of the IUT),

'Object Name = (any object name specified in the EPICS)

1. WAIT **Internal Processing Fail Time**
2. RECEIVE

DA = LOCAL BROADCAST | GLOBAL BROADCAST,

SA = IUT,

I-Have-Request,

'Device Identifier' = (the IUT's Device object),

'Object Identifier' = (the object identifier specified in the EPICS for this object),

'Object Name' = (the object name specified in step 1)

1. 使用Object Name，被测对象的设备实例号等于参数范围的下限

目的：当IUT的对象实例号等于参数下限且Object Name存在的情况下，验证IUT能否正确识别Who-Has服务请求。

测试步骤：

1. TRANSMIT

DA = LOCAL BROADCAST,

SA = TD,

Who-Has-Request,

'Device Instance Low Limit' = (The Device object instance number of the IUT),

'Device Instance High Limit' = (any value H: H > the Device object instance

number of the IUT),

'Object Name' = (any object name specified in the EPICS)

1. WAIT **Internal Processing Fail Time**
2. RECEIVE

DA = LOCAL BROADCAST | GLOBAL BROADCAST,

SA = IUT,

I-Have-Request,

'Device Identifier' = (the IUT's Device object),

'Object Identifier' = (the object identifier specified in step 1),

'Object Name' = (the object name specified in the EPICS for this object)

1. 使用Object Name，指向特定的MAC地址

目的：验证被测对象是否可以对非广播的Who-Has服务请求进行广播I-Have响应。

测试步骤：

1. TRANSMIT Who-Has-Request,

'Object Name' = (any object name specified in the EPICS),

1. WAIT **Internal Processing Fail Time**
2. RECEIVE

DA = LOCAL BROADCAST | GLOBAL BROADCAST,

SA =IUT,

I-Have-Request,

'Device Identifier' = (the IUT's Device object),

'Object Identifier' = (the object identifier specified in the EPICS for this object),

'Object Name' = (the object name specified in step 1)

1. 执行远程网络发起的Who-Has服务请求

此测试组的目的是验证远程网络发起的Who-Has请求服务过程能否正确执行。由于10.32.1中对异常Who-Has请求参数进行了全面的测试，因此本测试组中不进行相关测试。该组中的测试仅表示网络层寻址信息的变化。

1. 使用Object ID，从远程网络发起的全局广播

目的：验证IUT识别来自全局广播Who-Has服务请求以及正确响应的能力，并确保响应能够被请求的发起设备正确接收。

测试步骤：

1. TRANSMIT

DESTINATION = LOCAL BROADCAST,

SA = TD,

DNET = GLOBAL BROADCAST,

SNET = (any remote network number),

SADR = (any MAC address valid for the specified network),

Who-Has-Request,

'Object Identifier' = (any object identifier specified in the EPICS)

1. WAIT **Internal Processing Fail Time**
2. RECEIVE

DESTINATION = GLOBAL BROADCAST | REMOTE

BROADCAST (to the network specified in step 1),

I-Have-Request,

'Device Identifier' = (the IUT's Device object),

'Object Identifier' = (the object identifier specified in step 1),

'Object Name' = (the object name specified in the EPICS for this object)

1. 使用Object ID，远程广播

目的：验证IUT识别来自远程广播Who-Has服务请求以及正确响应的能力，并确保响应能够被请求的发起设备正确接收。

测试步骤：

1. TRANSMIT

DESTINATION = LOCAL BROADCAST,

SA = TD,

SNET = (any remote network number),

SADR = (any MAC address valid for the specified network),

Who-Has-Request,

'Object Identifier' = (any object identifier specified in the EPICS)

1. WAIT **Internal Processing Fail Time**
2. RECEIVE

DESTINATION = GLOBAL BROADCAST | REMOTE BROADCAST (to the

network specified in step 1),

I-Have-Request,

'Device Identifier' = (the IUT's Device object),

'Object Identifier' = (the object identifier specified in step 1),

'Object Name' = (the object name specified in the EPICS for this object)

* 1. Who-Is服务执行测试

此测试组的目的是验证Who-Is服务请求能否正确执行。

1. 执行本地网络发起的Who-Is服务请求

此测试组的目的是验证本地网络发起的Who-Is请求服务过程是否正确执行。

所需测试：无。

BACnet参考条款：16.10。

1. 本地广播，通用查询

目的：验证IUT是否可以正确响应不限制设备范围的本地广播Who-Is服务请求。

测试步骤：

1. TRANSMIT

DESTINATION = LOCAL BROADCAST,

Who-Is-Request

1. BEFORE **Unconfirmed Response Fail Time**

RECEIVE

DESTINATION = GLOBAL BROADCAST | LOCAL

BROADCAST | TD

I-Am-Request,

'I Am Device Identifier' = (the IUT's Device object),

'Max APDU Length Accepted' = (the value specified in the EPICS),

'Segmentation Supported' = (the value specified in the EPICS),

'Vendor Identifier' = (the identifier registered for this vendor)

1. 全局广播，通用查询

目的：验证IUT是否可以正确响应不限制设备范围的全局广播Who-Is服务请求。

测试步骤：

1. TRANSMIT

DESTINATION = GLOBAL BROADCAST,

Who-Is-Request

1. BEFORE **Unconfirmed Response Fail Time**

RECEIVE

DESTINATION = GLOBAL BROADCAST | LOCAL

BROADCAST | TD

I-Am-Request,

'I Am Device Identifier' = (the IUT's Device object),

'Max APDU Length Accepted' = (the value specified in the EPICS),

'Segmentation Supported' = (the value specified in the EPICS),

'Vendor Identifier' = (the identifier registered for this vendor)

1. 本地广播，IUT在设备范围之外的特定设备查询

目的：当IUT的设备实例号在查询的设备范围之外时，验证IUT能否忽略Who-Is请求。

测试步骤：

1. TRANSMIT

DESTINATION = LOCAL BROADCAST,

Who-Is-Request,

'Device Instance Range Low Limit'= (any value > 0 such that the Device object instance number

does not fall in the range between Device Instance Low

Limit and Device Instance High Limit),

'Device Instance Range High Limit' = (any value > Device Instance Low Limit such that the Device

object instance number does not fall in the range between

Device Instance Low Limit and Device Instance High Limit)

1. WAIT **Internal Processing Fail Time**
2. CHECK (verify that the IUT does not respond)
3. 本地广播，IUT的实例号等于设备范围下限的特定设备查询

目的：当IUT的设备实例号等于查询的设备范围下限时，验证IUT能否识别此Who-Is请求。

测试步骤：

1. TRANSMIT

DESTINATION = LOCAL BROADCAST,

Who-Is-Request,

'Device Instance Range Low Limit' = (The Device object instance number of the IUT),

'Device Instance Range High Limit' = (any value H: H > the Device object instance

number of the IUT)

1. BEFORE **Unconfirmed Response Fail Time**

RECEIVE

DESTINATION = GLOBAL BROADCAST | LOCAL

BROADCAST | TD

I-Am-Request,

'I Am Device Identifier' = (the IUT's Device object),

'Max APDU Length Accepted' = (the value specified in the EPICS),

'Segmentation Supported' = (the value specified in the EPICS),

'Vendor Identifier' = (the identifier registered for this vendor)

1. 本地广播，IUT的实例号等于设备范围上限的特定设备查询

目的：当IUT的设备实例号等于查询的设备范围上限时，验证IUT能否识别此Who-Is请求。

测试步骤：

1. TRANSMIT

DESTINATION = LOCAL BROADCAST,

Who-Is-Request,

'Device Instance Range Low Limit' = (any value L: 0 ≤ L < the Device object instance

number of the IUT),

'Device Instance Range High Limit' = (the Device object instance number of the IUT)

1. BEFORE **Unconfirmed Response Fail Time**

RECEIVE

DESTINATION = GLOBAL BROADCAST | LOCAL

BROADCAST | TD

I-Am-Request,

'I Am Device Identifier' = (the IUT's Device object),

'Max APDU Length Accepted' = (the value specified in the EPICS),

'Segmentation Supported' = (the value specified in the EPICS),

'Vendor Identifier' = (the identifier registered for this vendor)

1. 本地广播，IUT的实例号在设备范围内的特定设备查询

目的：当IUT的设备实例号在查询的设备范围内时，验证IUT能否识别此Who-Is请求。

测试步骤：

1. TRANSMIT

DESTINATION = LOCAL BROADCAST,

Who-Is-Request,

'Device Instance Range Low Limit' = (any value L: 0 ≤ L < the Device object instance

number of the IUT),

'Device Instance Range High Limit' = (any value H: H > the Device object instance

number of the IUT)

1. BEFORE **Unconfirmed Response Fail Time**

RECEIVE

DESTINATION = GLOBAL BROADCAST | LOCAL

BROADCAST | TD

I-Am-Request,

'I Am Device Identifier' = (the IUT's Device object),

'Max APDU Length Accepted' = (the value specified in the EPICS),

'Segmentation Supported' = (the value specified in the EPICS),

'Vendor Identifier' = (the identifier registered for this vendor)

1. 执行远程网络发起的Who-Is服务请求

此测试组的目的是验证远程网络发起的Who-Is请求服务过程能否正确执行。在10.33.1小节中对各种参数变异的Who-Is请求进行了全面的测试，因此本测试组中不进行相关的测试。本测试组中的测试仅表示网络层寻址信息的变化。

1. 来自远程网络的全局广播通用查询

目的：验证IUT识别来自远程全局广播Who-Is服务请求以及正确响应的能力，并确保响应能够被请求的发起设备正确接收。

测试步骤：

1. TRANSMIT

DESTINATION = GLOBAL BROADCAST,

SNET = (any remote network number),

SADR = (any MAC address valid for the specified network),

Who-Is-Request

1. BEFORE **Unconfirmed Response Fail Time**

RECEIVE

DESTINATION = GLOBAL BROADCAST | REMOTE BROADCAST (to the network specified y SNET in step 1) | TD,

I-Am-Request,

'I Am Device Identifier' = (the IUT's Device object),

'Max APDU Length Accepted' = (the value specified in the EPICS),

'Segmentation Supported' = (the value specified in the EPICS),

'Vendor Identifier' = (the identifier registered for this vendor)

1. 远程广播的通用查询

目的：验证IUT识别远程广播Who-Is服务请求以及正确响应的能力，并确保响应能够被请求的发起设备正确接收。

测试步骤：

1. TRANSMIT

DESTINATION = LOCAL BROADCAST,

SNET = (any remote network number),

SADR = (any MAC address valid for the specified network),

Who-Is-Request

1. BEFORE **Unconfirmed Response Fail Time**

RECEIVE

DESTINATION = GLOBAL BROADCAST | REMOTE BROADCAST (to the network specified by SNET in step 1) | TD,

I-Am-Request,

'I Am Device Identifier' = (the IUT's Device object),

'Max APDU Length Accepted' = (the value specified in the EPICS),

'Segmentation Supported' = (the value specified in the EPICS),

'Vendor Identifier' = (the identifier registered for this vendor)

1. 指向远程设备的通用查询

目的：验证IUT能否采用全局广播、远程广播或单播形式的I-Am服务进行响应。

测试步骤：

1. TRANSMIT

DESTINATION = IUT,

SNET = (any remote network number),

SADR = (any MAC address valid for the specified network),

Who-Is-Request

1. BEFORE **Unconfirmed Response Fail Time**

RECEIVE

DESTINATION = GLOBAL BROADCAST | LOCAL

BROADCAST

I-Am-Request,

'I Am Device Identifier' = (the IUT's Device object),

'Max APDU Length Accepted' = (the value specified in the EPICS),

'Segmentation Supported' = (the value specified in the EPICS),

'Vendor Identifier' = (the identifier registered for this vendor)

* 1. VT-Open服务执行测试

本节定义了验证支持执行VT-Open服务请求所必需的测试。

所需测试：执行ReadProperty服务测试，10.18。

BACnet参考条款：17.2。

测试思想：尝试打开IUT支持的每个终端类型建立VT会话。读取Device对象的Active\_VT\_Sessions属性确认会话已打开。VT数据的交换在10.36中完成。除非IUT无法支持多打开个会话，否则VT会话将保持打开状态。这些创建的会话可用于在10.35测试。

1. VT-class为缺省终端类型

目的：验证IUT是否响应缺省终端类型的VT-Open请求以建立会话，并且VT会话在IUTDevice对象的Active\_VT\_Sessions属性中得到反映。

测试步骤：

1. TRANSMIT VT-Open-Request,

'VT-class' = DEFAULT\_TERMINAL,

'Local VT Session Identifier' = (any valid unique session identifier)

1. RECEIVE VT-Open-ACK,

'Remote VT Session Identifier' = (any valid unique session identifier)

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = (the IUT’s Device object),

'Property Identifier' = Active\_VT\_Sessions

1. RECEIVE ReadProperty-ACK,

'Object Identifier' = (the IUT’s Device object),

'Property Identifier' = Active\_VT\_Sessions,

'Property Value' = (any list of sessions that contains the session ID

pair from steps 1 and 2)

1. IF(the IUT can have only one open VT-session)THEN{

TRANSMIT VT-Close-Request,

'List of Remote VT Session Identifiers' = (the remote session identifier from step 2)

ECEIVE BACnet-Simple-ACK-PDU

}

注意事项：成功完成第1步到第4步时结果为通过。如果步骤5失败，则表示VT-Close服务失败。

1. 其他VT-classes

目的：验证IUT是否响应所有支持的可选VT类的VT-Open请求，以及VT会话是否反映在IUT的Device对象的Active\_VT\_Sessions属性中。如果DEFAULT\_TERMINAL是唯一支持的VT类，则省略该测试。

测试步骤：

1. REPEAT X = (the supported optional VT-classes) DO{

TRANSMIT VT-Open-Request,

'VT-class' = X,

'Local VT Session Identifier' = (any valid unique session identifier)

RECEIVE VT-Open-ACK,

'Remote VT Session Identifier' = (any valid unique session identifier)

TRANSMIT ReadProperty-Request,

'Object Identifier' = (IUT的Device object),

'Property Identifier' = Active\_VT\_Sessions

Receive ReadProperty-ACK,

'Object Identifier' = (IUT的Device object),

'Property Identifier' = Active\_VT\_Sessions,

'Property Value' = (any list of sessions that contains the session ID

pair created above)

IF (the maximum number of open VT-sessions has been reached) THEN {

TRANSMIT VT-Close-Request,

'List of Remote VT Session Identifiers' = (any active remote session identifier)

RECEIVE BACnet-Simple-ACK-PDU

}

}

* 1. VT-Close服务执行测试

本节定义了验证支持执行VT-Close服务请求所必需的测试。

所需测试：执行ReadProperty服务测试，10.18; 执行VT-Open服务测试，10.34。

BACnet参考条款：17.3。

1. 从多个打开的VT会话中关闭一个

目的：验证IUT在多个会话打开时响应VT-Close请求以终止单个VT会话的能力。如果IUT不支持打开多个VT会话，则省略该测试。

配置要求：IUT应配置打开多个VT会话。10.34测试可用于实现此目的。

测试步骤：

1. TRANSMIT VT-Close-Request,

'List of Remote VT Session Identifiers' = (any single identifier for an open session identifier)

1. RECEIVE BACnet-Simple-ACK-PDU
2. TRANSMIT ReadProperty-Request,

'Object Identifier' = (IUTDevice object),

'Property Identifier' = Active\_VT\_Sessions

1. RECEIVE ReadProperty-ACK,

'Object Identifier' = (IUTDevice object),

'Property Identifier' = Active\_VT\_Sessions,

'Property Value' = (any list of sessions that does not contain the

closed session)

1. 关闭多个打开的VT会话

目的：验证IUT是否响应VT-Close请求以终止多个打开的VT会话。如果IUT不支持打开多个VT会话，则应省略该测试。

配置要求：IUT应配置打开多个VT会话。10.34测试可用于实现此目的。

测试步骤：

1. TRANSMIT VT-Close-Request,

'List of Remote VT Session Identifiers' = (at least two session identifiers corresponding to

open sessions)

1. RECEIVE BACnet-Simple-ACK-PDU
2. TRANSMIT ReadProperty-Request,

'Object Identifier' = (the IUT’s Device object),

'Property Identifier' = Active\_VT\_Sessions

1. RECEIVE ReadProperty-ACK,

'Object Identifier' = (the IUT’s Device object),

'Property Identifier' = Active\_VT\_Sessions,

'Property Value' = (any list of sessions that does not contain the

closed sessions)

1. 关闭单个已打开的VT会话

目的：验证IUT是否响应VT-Close请求以终止单个已打开的VT会话。

配置要求：IUT应配置打开一个VT会话。10.34测试可用于实现此目的。

测试步骤：

1. TRANSMIT VT-Close-Request,

'List of Remote VT Session Identifiers' = (the session identifier corresponding to the open session)

1. RECEIVE BACnet-Simple-ACK-PDU
2. TRANSMIT ReadProperty-Request,

'Object Identifier' = (the IUT’s Device object),

'Property Identifier' = Active\_VT\_Sessions

1. RECEIVE ReadProperty-ACK,

'Object Identifier' = (the IUT’s Device object),

'Property Identifier' = Active\_VT\_Sessions,

'Property Value' = (an empty list)

* 1. VT-Data服务执行测试

使用VT-Data服务交换VT数据需要启动和执行服务。通过8.38测试足以证明VT-Data服务的执行。

* 1. 请求秘钥(RequestKey)服务执行测试

所需测试：AddListElement服务执行测试，10.14; AddListElement Service服务执行测试，9.14。

BACnet参考条款：24.2.1和24.4。

目的：验证密钥服务器正确执行请求秘钥服务请求的能力。如果IUT不是密钥服务器，则应省略该测试。

测试思想：请求RequestKey服务过程规定了用于建立加密会话密钥的一系列步骤。该测试基于两个设备（客户端设备A和服务器设备B）的存在。客户端A希望向服务器B发送安全服务请求。在此之前，必须建立会话密钥。如果秘钥服务器(IUT)能够正确执行请求秘钥服务，则此测试由一系列相关的交互报文完成。

IUT在此过程中扮演密钥服务器的角色。使用两个单独的具有密钥服务器已知的私钥的测试设备或者一个具有两个单独的密钥和两个唯一的Device对象标识符的测试设备，分别扮演设备A和设备B的角色。

实际上，要求IUT知道设备A和B的最大接收APDU长度，以便根据BACnet条款24.1.4的要求填充其加密的APDU。IUT的最大接收APDU长度可以在此测试期间通过BACnet服务获取。在测试描述中未指出实现相关功能的PDU。

本测试中的所有PDU首先按照BACnet条款24.1.4进行填充，然后使用指定密钥按照BACnet条款24.4进行加密。

配置要求：IUT应配置私有56位加密密钥(PKA和PKB)，设备对象标识符以及与设备A和设备B对应的MAC地址。

测试步骤：

1. TRANSMIT RequestKey-Request,

DESTINATION = IUT,

SOURCE = (device A),

'Requesting Device Identifier' = (the Device object identifier for device A),

'Requesting Device Address' = (a BACnetAddress for device A),

'Remote Device Identifier' = (the Device object identifier for device B)

'Remote Device Address' = (a BACnetAddress for device B)

注意：该PDU的服务请求部分应使用PKA加密。

1. BEFORE **Internal Processing Fail Time**

RECEIVE

DESTINATION = (device A),

SOURCE = IUT,

Authenticate-Request,

'Pseudo Random Number' = (any valid pseudo random number),

'Expected Invoke ID' = (the invoke ID used in step 1)

注意：该PDU的服务请求部分应使用PKA加密。

1. TRANSMIT

DESTINATION = IUT,

SOURCE = (device A),

Authenticate-Request-ACK,

'Modified Random Number' = (the modified pseudo random number)

注意：该PDU的服务请求部分应使用PKA加密。至此，步骤1中的请求已经过身份验证，并且请求RequestKey服务过程仍在继续。

1. BEFORE **Internal Processing Fail Time**

RECEIVE

DESTINATION = (device B),

SOURCE = IUT,

AddListElement-Request,

'Object Identifier' = (the Device object identifier for device B),

'Property Identifier' = List\_Of\_Session\_Keys,

'List of Elements' = (SKAB, BACnetAddress for device A)

注意：List\_Of\_Session\_Keys这个属性值应使用PKB加密。

1. TRANSMIT

DESTINATION = IUT,

SOURCE =(device B),

Authenticate-Request,

'Pseudo Random Number' = (any valid pseudo random number),

'Expected Invoke ID' = (the invoke ID used in step 6)

注意：该PDU的服务请求部分应使用PKB加密。

1. BEFORE **Acknowledgment Fail Time**

RECEIVE

DESTINATION = (device B),

SOURCE = IUT,

Authenticate-Request-ACK,

'Modified Random Number' = (the modified pseudo random number)

注意：该PDU的服务请求部分应使用PKB加密。至此，步骤4中的请求已经过身份验证。

1. TRANSMIT

DESTINATION = IUT,

SOURCE = (the modified pseudo random number),

BACnet-Simple-ACK-PDU

注意：这是对步骤4中成功执行AddListElement请求的确认。

1. BEFORE **Internal Processing Fail Time**

RECEIVE

DESTINATION = (device A),

SOURCE = IUT,

AddListElement-Request,

'Object Identifier' = (the Device object identifier for device A),

'Property Identifier' = List\_Of\_Session\_Keys,

'List of Elements' = (SKAB, BACnetAddress for device B)

注意：List\_Of\_Session\_Keys这个属性值应使用PKA加密。

1. TRANSMIT

DESTINATION = IUT,

SOURCE =(device A),

Authenticate-Request,

'Pseudo Random Number' = (any valid pseudo random number),

'Expected Invoke ID' = (the invoke ID used in step 8)

注意：该PDU的服务请求部分应使用PKA加密。

1. BEFORE **Acknowledgment Fail Time**

RECEIVE

DESTINATION = (device A),

SOURCE = IUT,

Authenticate-Request-ACK,

'Modified Random Number' = (the modified pseudo random number)

注意：该PDU的服务请求部分应使用PKA加密。至此，步骤12中的请求已经过身份验证。

1. TRANSMIT

DESTINATION = IUT,

SOURCE = (device A),

BACnet-Simple-ACK-PDU

注意：这是对步骤8中成功执行AddListElement请求的确认。

1. BEFORE **Acknowledgment Fail Time**

RECEIVE

DESTINATION = (device A),

SOURCE = IUT,

BACnet-Simple-ACK-PDU

注意：这是对步骤1中成功执行请求RequestKey服务请求的确认。

* 1. 认证（Authenticate）服务执行测试

本节定义了验证支持执行Authenticate服务请求所必需的测试。

本节中的测试要求IUT知道测试设备的最大接收APDU长度，以便可以按照BACnet条款24.1.4的要求填充其加密的APDU。IUT的最大接收APDU长度可以在此测试期间通过BACnet服务获取。在测试描述中未指出实现相关功能的PDU。

本测试中的所有PDU首先按照BACnet条款24.1.4进行填充，然后使用指定密钥按照BACnet条款24.4进行加密。

1. 建立会话密钥

所需测试：AddListElement服务执行10.14。

BACnet参考条款：24.2.1和24.5。

目的：本节中的后续测试要求在测试执行之前建立安全会话。此测试用于验证IUT是否可以对尝试建立此类会话的请求进行响应。如果IUT只能在配置时建立会话，则该测试应省略。

测试思想：测试设备用作密钥服务器并尝试将会话密钥传递给IUT。IUT验证实际上是由测试设备提供的密钥。

配置要求：IUT应配置专用的56位加密密钥。

测试步骤：

1. TRANSMIT AddListElement-Request,

'Object Identifier' = (the Device object identifier of the IUT),

'Property Identifier' = List\_Of\_Session\_Keys,

'List of Elements' = (SKTD,IUT, BACnetAddress for TD)

注意：List of Elements应使用PKIUT加密。

1. BEFORE **Internal Processing Fail Time**

RECEIVE

Authenticate-Request,

'Pseudo Random Number' = (any valid pseudo random number),

'Expected Invoke ID' = (the invoke ID used in step 1)

注意：该PDU的服务请求部分应使用PKIUT加密。

1. TRANSMIT Authenticate-Request-ACK,

'Modified Random Number' = (the modified pseudo random number)

注意：至此步骤1的请求已经过验证。

1. BEFORE **Acknowledgment Fail Time**

RECEIVE BACnet-Simple-ACK-PDU

注意：这是对步骤1中成功执行AddListElement请求的确认。

1. 端认证

所需测试：无。

BACnet参考条款：24.2.2和24.5。

目的：验证设备正确执行Authentication服务请求以实现端认证的能力。如果IUT只是一个秘钥服务器，则应省略该测试。

配置要求：IUT应配置专用的56位加密密钥。测试设备和IUT之间使用的会话密钥SKTD，IUT应该预先建立。建立过程可以通过执行10.38.1中的测试来完成。

测试步骤：

1. TRANSMIT Authenticate-Request,

'Pseudo Random Number' = (any valid pseudo random number)

2. BEFORE Acknowledgment Fail Time

RECEIVE Authenticate-Request-ACK,

'Modified Random Number' = (the modified pseudo random number)

1. 报文执行认证

所需测试：ReadProperty服务执行测试，10.18。

BACnet参考条款：24.2.3和24.5。

目的：验证设备是否具有正确执行Authenticate服务请求以实现报文执行认证的能力。如果IUT只是一个秘钥服务器，则应省略该测试。

测试思想：测试设备和IUT之间的安全会话已经建立。测试设备使用经过认证的服务执行ReadProperty请求的过程。

配置要求：IUT应配置专用的56位加密密钥。测试设备和IUT之间使用会话密钥SKTD，IUT应该预先建立。建立过程可以通过执行10.38.1中的测试来完成。

测试步骤：

1. TRANSMIT Authenticate-Request,

'Pseudo Random Number' = (any valid pseudo random number),

'Expected Invoke ID' = (any valid invoke ID)

注意：该PDU的服务请求部分应使用SKTD，IUT加密。

1. TRANSMIT ReadProperty-Request,

Invoke ID = (the 'Expected Invoke ID' used in step 1),

'Object Identifier' = (any object supported by the IUT),

'Property Identifier' = (any supported property of the specified object)

1. BEFORE **Internal Processing Delay Time**

RECEIVE Authenticate-Request-ACK,

'Modified Random Number' =(the modified pseudo random number)

注意：该PDU的服务请求部分应使用SKTD，IUT加密。

1. BEFORE **Internal Processing Delay Time**

RECEIVE ReadProperty-ACK,

'Object Identifier' = (the object identifier used in step 2),

'Property Identifier' = (the property identifier used in step 2),

'Property Value' = (the value of the property as specified in the EPICS)

1. 报文发起认证

本节定义了验证支持执行消息发起认证的Authenticate服务所必须的测试。

1. 作为密钥服务器的消息发起认证

9.37涵盖了作为密钥服务器执行消息发起认证的测试。

1. 端到端的消息发起认证

所需测试：ReadProperty服务发起测试，9.18。

BACnet参考条款：24.2.4和24.5。

目的：验证以实现消息发起认证为目的正确执行Authenticate服务的能力。如果IUT仅是密钥服务器，则省略该测试。

测试思想：测试设备和IUT之间已建立安全会话。IUT发起针对测试设备的应用服务请求，然后测试设备遵循验证请求发起源的程序进行处理。

配置要求：IUT应配置专用的56位加密密钥。要预先建立测试设备和IUT之间使用会话密钥SKTD，IUT的安全会话。这可以通过执行10.38.1中的测试来完成。应提供一种方法使IUT发起ReadProperty服务请求。

测试步骤：

1. MAKE (the IUT initiate a ReadProperty service request)
2. RECEIVE ReadProperty-Request,

'Object Identifier' = (any standard object),

'Property Identifier' = (any property of the specified object)

1. TRANSMIT Authenticate-Request

'Pseudo Random Number' = (any valid pseudo random number),

'Expected Invoke ID' = (the invoke ID used in step 2)

注意：该PDU的服务请求部分应使用SKTD，IUT加密。

1. BEFORE **Acknowledgment Fail Time**

RECEIVE

Authenticate-Request-ACK,

'Modified Random Number' = (the modified pseudo random number)

注意：该PDU的服务请求部分应使用SKTD，IUT加密。

1. 操作员认证

本节定义了验证支持执行以操作员认证为目的的Authenticate服务所必须的测试。如果IUT不是执行操作员认证的密钥服务器，则省略这些测试。

测试思想：测试设备发送加密请求以验证操作员，然后IUT遵循验证操作员的程序进行处理。

配置要求：IUT应配置私有的56位加密密钥PKTD，它是对应于测试设备和已知的操作员名称/密码组合的秘钥。

1. 接受登录

所需测试：无。

BACnet参考条款：24.2.5和24.5。

目的：在预期认证成功的情况下，验证以实现操作员身份为目的的验证密钥服务器正确执行Authenticate服务请求的能力。

测试步骤：

1. TRANSMIT Authenticate-Request,

'Pseudo Random Number' = (any valid pseudo random number),

'Operator Name' = (the operator name configured for this test),

'Operator Password' = (the operator password configured for this test)

注意：该PDU的服务请求部分应使用PKTD加密。

1. BEFORE **Acknowledgment Fail Time**

RECEIVE Authenticate-ACK,

'Modified Random Number' = (the modified pseudo random number)

注意：该PDU的服务请求部分应使用PKTD加密。

1. 拒绝登录

所需测试：无。

BACnet参考条款：24.2.5和24.5。

目的：验证密钥服务器在预期验证失败的情况下正确执行以实现操作员验证为目的的Authenticate服务请求的能力。

测试步骤：

1. TRANSMIT Authenticate-Request,

'Pseudo Random Number' = (any valid pseudo random number),

'Operator Name' = (the operator name configured for this test),

'Operator Password' = (a password other than the one configured for this est)

注意：该PDU的服务请求部分应使用PKTD加密。

1. BEFORE **Acknowledgment Fail Time**

RECEIVE BACnet-Error-PDU,

Error Class = SECURITY,

Error Code = PASSWORD\_FAILURE

1. 加密会话

所需测试： ReadProperty服务执行测试，10.18.

BACnet参考条款：24.3.1，24.3.2，和24.5

目的：验证IUT正确执行Authenticate请求来启动和终止加密会话

测试思想：TD试图通过发送一个Authenticate请求对IUT启动一个加密会话。IUT依照BACnet 24.3.1的流程验证请求的身份并启动一个会话。接下来，TD从IUT的设备对象里读出List\_Of\_Session\_Keys属性。然后TD通过发送另一个Authenticate请求终止该会话。IUT通过实施BACnet 24.3.2中的过程对编码的加密会话进行响应。TD再次读取List\_Of\_Session\_Keys，从而导致错误响应。注意，此测试中所有消息的服务请求部分都是使用SKTD，IUT加密的。

配置要求：用一个专有56位加密密钥配置IUT。在TD和IUT之间，通过使用一种会话密钥SKTD,IUT建立安全会话，这可以通过执行10.38.1中的测试来实现。

测试步骤：

1. TRANSMIT Authenticate-Request,

'Pseudo Random Number' = (any valid pseudo random number),

'Start Enciphered Session' = TRUE

1. BEFORE **Internal Processing Fail Time**

RECEIVE Authenticate-Request,

'Pseudo Random Number' = (any valid pseudo random number),

'Expected Invoke ID' = (the invoke ID used in step 1)

1. TRANSMIT Authenticate-ACK,

'Modified Random Number' = (the modified random number from step 2)

1. BEFORE **Acknowledgment Fail Time**

RECEIVE Authenticate-ACK,

'Modified Random Number' = (the modified random number from step1)

注意：这时候已启动加密会话。

1. VERIFY (the IUT's Device object), List\_Of\_Session\_Keys = (a list containing the session key for this session)
2. TRANSMIT Authenticate-Request,

'Pseudo Random Number' = (any valid pseudo random number),

'Start Enciphered Session' = FALSE

1. BEFORE **Internal Processing Fail Time**

RECEIVE Authenticate-Request,

'Pseudo Random Number' = (any valid pseudo random number),

'Expected Invoke ID' = (the invoke ID used ins step 6)

1. TRANSMIT Authenticate-ACK,

'Modified Random Number' = (the modified random number from step 7)

1. BEFORE **Acknowledgment Fail Time**

RECEIVE Authenticate-ACK,

'Modified Random Number' = (the modified random number from step 6)

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = (the Device object of the IUT),

'Property Identifier' = List\_Of\_Session\_Keys

1. BEFORE **Acknowledgment Fail Time**

RECEIVE BACnet-Error-PDU,

Error Class = SECURITY,

Error Code = OTHER

* 1. 服务执行通用测试

这一测试部分定义了证明一个设备可与BACnet网络上共存所必需的测试。这些是不与任何特定网络服务关联的常规测试。

1. 不支持的确认服务测试

所需测试：无

BACnet参考条款：UNRECOGNIZED\_SERVICE,18.9.9

目的：该测试用例验证了IUT将拒绝其不支持的任何已确认服务。

测试步骤：

1. REPEAT X = (all confirmed services that the IUT does not execute)DO {

TRANSMIT X

RECEIVE BACnet-Reject-PDU,

'Reject Reason' = UNRECOGNIZED\_SERVICE

}

2. TRANSMIT (a currently undefined confirmed service)

3. RECEIVE BACnet-Reject-PDU,

'Reject Reason' = UNRECOGNIZED\_SERVICE

传递结果：该设备正确响应每一个不支持的确认服务。

1. 不支持的未确认服务测试

所需测试：无

目的：此测试用例验证IUT将接受并丢弃它不支持的所有未确认服务。在确定要发送到IUT的服务集时，无论IUT是否支持，都应包括UnconfirmedPrivateTransfer服务。UnconfirmedPrivateTransfer服务应与设备不支持的供应商ID/服务编号对一起发送。

配置需求：要求IUT处于正常工作状态，不触发任何请求。

测试步骤：

1. VERIFY System\_Status == OPERATIONAL | OPERATIONAL\_READ\_ONLY

2. REPEAT X = (all unconfirmed services that the IUT does not execute) DO {

TRANSMIT X

BEFORE **Internal Processing Fail Time**

CHECK (verify that the IUT did not reset and that the IUT did not send any packets)

VERIFY System\_Status = (the value of System\_Status read in step 1)

}

传输结果：作为该服务响应，IUT没有复位或发数据包。

1. 网络层协议测试
   1. 处理来自远程网络的应用层信息

所需测试：ReadProperty服务执行测试，10.18

BACnet参考条款：6.5.4

目的：验证IUT能够响应来自远程网络的请求。

测试思想：TD发送一个ReadProperty请求消息，其中包含的网络层信息表明其来自远程网络。来自IUT的响应包括正确的DNET和DADR信息，以便此消息能够到达原始请求者。MAC层目标地址可以是一个广播，表示IUT无法知道路由器的地址，也可以是相关路由器的MAC地址。

测试步骤：

1. TRANSMIT

DESTINATION = IUT,

SOURCE = TD,

SNET = (any network number that is not the local network),

SADR = (any valid MAC address consistent with the source network),

ReadProperty-Request,

'Object Identifier' = (any supported object),

'Property Identifier' = (any required property of the specified object)

1. RECEIVE

DESTINATION = LOCAL BROADCAST | (an appropriate router address),

SOURCE = IUT,

DNET = (the SNET specified in step 1),

DADR = (the SADR specified in step 1),

Hop Count = 255,

ReadProperty-ACK,

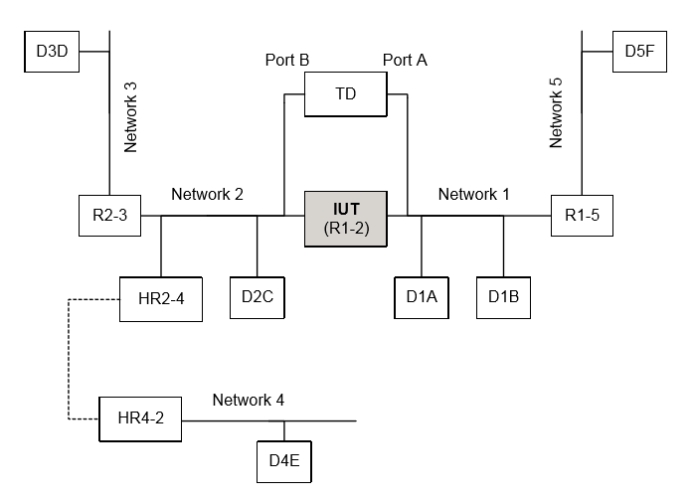
'Object Identifier' = (the object specified in step 1),

'Property Identifier' = (the property specified in step 1),

'Property Value' = (the value of the specified property as defined in the EPICS)

* 1. 路由器功能测试

本条款定义了为证明BACnet路由器功能所必须的测试。假定路由器有2端口，端口1直接连接网络1，端口2直接连接网络2。如果路由器有多于2个端口，对于每种可能的两个端口的组合，应都使用以下步骤对进行测试。用于这些测试的网络配置逻辑如图11-1所示。本节中的测试说明假设TD可以同时连接到网络1和网络2，并且可以模拟所有其他设备。与网络1的连接称为“Port A”，与网络2的连接称为“Port B”。可接受的替代方法是使用所示的真实设备构建互联网络。逻辑网络3和5必须使用不同的LAN技术，这两者都不同于网络1和2，以确保路由器可以支持具有不同大小MAC地址的远程网络。



**图11-1** 路由器功能测试的网络配置逻辑

因特网中所包含的逻辑设备如下：

IUT: 待测设备，网络1与网络2之间的一个路由器

D1A: 网络1上的设备

D1B: 网络1上的设备

D2C: 网络2上的设备

D3D: 网络3上的设备

D4E: 网络4上的设备

D5F: 网络5上的设备

R1-5: 网络1与网络5之间的路由器

R2-3: 网络2与网络3之间的路由器

HR2-4: 直接连接到网络2的半路由器，能与连接到网络4的半路由器H4-2产生一个PTP连接

配置要求：IUT必须配置有路由表，这些路由表指示网络1直接连接到端口1，网络2直接连接到端口2，如图11-1所示。路由表不应包含其他内容。

路由器功能测试应按照本标准中定义的顺序进行。在某些情况下，成功完成测试用例要求IUT从上一个测试的最终状态开始。

1. 开始

BACnet参考条款：6.6.2

目的：验证IUT开机时广播一个相应的I-Am-Router-To-Network消息

测试步骤：

1. MAKE (power cycle the router to make it reinitialize)
2. RECEIVE PORT A,

DESTINATION = LOCAL BROADCAST,

I-Am-Router-To-Network,

Network Numbers = 2

1. RECEIVE PORT B,

DESTINATION = LOCAL BROADCAST,

I-Am-Router-To-Network,

Network Numbers = 1

1. 处理网络层消息

本条款定义的测试用来验证IUT可正确处理网络层消息。

1. 前向 I-Am-Router-To-Network

BACnet参考条款：6.6.3.3

目的：验证当IUT从另一个路由器接收到一条I-Am-Router-To-Network消息时，将会广播一条相应的I-Am-Router-To-Network消息。

测试思想：通过发送I-Am-Router-To-Network消息，TD同时触发路由器R2-3和R1-5，IUT将会更新其路由表(在11.2.2.6.1验证)并向除了接收到I-Am-Router-To-Network消息的端口之外的所有端口上发送I-Am-Router-To-Network消息。

测试步骤：

1. TRANSMIT PORT B

DESTINATION=LOCAL BROADCAST,

I-Am-Router-To-Network，

Network Numbers=3

1. RECEIVE PORT A,

DESTINATION = LOCAL BROADCAST,

I-Am-Router-To-Network,

Network Numbers = 3

1. TRANSMIT PORT A,

DESTINATION = LOCAL BROADCAST,

I-Am-Router-To-Network,

Network Numbers = 5

1. RECEIVE PORT B,

DESTINATION = LOCAL BROADCAST,

I-Am-Router-To-Network,

Network Numbers = 5

1. 执行Who-Is-Router-To-Network

所需测试：11.2.2.1

BACnet参考条款：6.6.3.2和6.6.3.3

1. 没有指定的网络号

目的：验证当IUT接收到没有指定网络号的“Who-Is-Router-To-Network”消息时，它将广播列出所有下游网络的I-AM-Router-to-Network消息。

测试步骤：

1. TRANSMIT PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = D1A,

Who-Is-Router-to-Network

1. RECEIVE PORT A,

DESTINATION = LOCAL BROADCAST,

I-Am-Router-To-Network,

Network Numbers = 2, 3 | 3, 2

1. TRANSMIT PORT B,

DESTINATION = LOCAL BROADCAST,

SOURCE = D2C,

Who-Is-Router-to-Network

1. RECEIVE PORT B,

DESTINATION = LOCAL BROADCAST,

I-Am-Router-To-Network,

Network Numbers = 1, 5 | 5, 1

1. 指定已知的远程网络号

目的：验证当IUT接收到路由表中包含的具有指定网络号的“Who-Is-Router-To-Network”消息时，它将广播I-Am-Router-to-Network消息。

测试步骤：

1. TRANSMIT PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = D1A,

Who-Is-Router-To-Network,

Network Number = 2

1. RECEIVE PORT A,

DESTINATION = LOCAL BROADCAST,

I-Am-Router-To-Network,

Network Numbers = 2

1. TRANSMIT PORT B,

DESTINATION = LOCAL BROADCAST,

SOURCE = D2C

Who-Is-Router-To-Network,

Network Number = 5

1. RECEIVE PORT B,

DESTINATION = LOCAL BROADCAST,

I-Am-Router-To-Network,

Network Numbers = 5

1. 指定网络号且路由器不响应

目的：要验证IUT是否收到Who-Is-Router-To-Network消息（指定了可通过接收I-Am-Router-To-Network消息的同一端口可达的网络的网络号），没有响应。

测试步骤：

1. TRANSMIT PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = D1A,

Who-Is-Router-To-Network,

Network Number = 1

1. Wait **Internal Processing Fail Time**
2. CHECK (verify that the IUT does not respond)
3. TRANSMIT PORT B,

DESTINATION = LOCAL BROADCAST,

SOURCE = D2C,

Who-Is-Router-To-Network,

Network Number = 3

1. Wait **Internal Processing Fail Time**
2. CHECK(verify that the IUT does not respond)
3. 指定未知和无法访问的网络号

目的：验证如果IUT接收到指定未知网络号的Who-Is-Router-To-Network消息，它将尝试定位网络。

测试步骤：

1. TRANSMIT PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = D1A,

Who-Is-Router-To-Network,

Network Number = 4

1. RECEIVE PORT B,

DESTINATION = LOCAL BROADCAST,

SA = IUT,

SNET = 1,

SADR = D1A,

Who-Is-Router-To-Network,

Network Number = 4

1. CHECK (verify that no I-Am-Router-To-Network is transmitted on Port A)
2. TRANSMIT PORT B,

DESTINATION = LOCAL BROADCAST,

SOURCE = D2C,

Who-Is-Router-To-Network,

Network Number = 4

1. RECEIVE PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = IUT,

SNET = 2,

SADR = D2C,

Who-Is-Router-To-Network,

Network Number = 4

1. CHECK (verify that no I-Am-Router-To-Network is transmitted on Port B)
2. 发现未知网络

目的：为了验证在搜索并发现通往未知网络的路径之后，IUT会发送相应的I-Am-Router-To-Network消息。

测试步骤：

1. TRANSMIT PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = D1A,

Who-Is-Router-To-Network,

Network Number = 6

1. RECEIVE PORT B,

DESTINATION = LOCAL BROADCAST,

SA = IUT,

SNET = 1,

SADR = D1A,

Who-Is-Router-To-Network,

Network Number = 6

1. TRANSMIT PORT B,

DESTINATION = LOCAL BROADCAST,

SOURCE = R2-3,

I-Am-Router-To-Network,

Network Numbers = 6

1. RECEIVE PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = IUT,

I-Am-Router-To-Network,

Network Numbers = 6

1. 从一个远程网络转发Who-Is-Router-To-Network

目的：验证如果IUT收到一条指定未知网络号和包含网络层源地址信息的Who-Is-Router-To-Network消息时，能转发这条Who-Is-Router-To-Network消息。

测试步骤：

1. TRANSMIT PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = R1-5,

SNET = 5,

SADR = D5F,

Who-Is-Router-To-Network,

Network Number = 4

1. RECEIVE PORT B,

DESTINATION = LOCAL BROADCAST,

SOURCE = IUT

SNET = 5,

SADR = D5F,

Who-Is-Router-To-Network,

Network Number = 4

1. CHECK (verify that no I-Am-Router-To-Network is transmitted on Port A)
2. TRANSMIT PORT B,

DESTINATION = LOCAL BROADCAST,

SOURCE = R2-3,

SNET = 3,

SADR = D3D,

Who-Is-Router-To-Network,

Network Number = 4

1. RECEIVE PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = IUT,

SNET = 3,

SADR = D3D,

Who-Is-Router-To-Network,

Network Number = 4

1. CHECK (verify that no I-Am-Router-To-Network is transmitted on Port B)
2. 转发I-Could-Be-Router-To-Network

BACnet参考条款：6.6.3.4

目的：验证IUT收到一条I-Could-Be-Router-To-Network消息后将会转发给目标接收者

测试步骤：

1. TRANSMIT PORT B,

DA = IUT,

SOURCE = HR2-4,

DNET = 1,

DADR = D1A,

Hop Count = 255,

I-Could-Be-Router-To-Network,

Network Number = 4,

Performance Index = 6

1. RECEIVE PORT A,

DA = D1A,

SA = IUT,

SNET = 2,

SADR = HR2-4,

I-Could-Be-Router-To-Network,

Network Number = 4,

Performance Index = 6

Router-Busy-To-Network

1. Router-Busy-To-Network

BACnet参考条款：6.6.3.6

1. 向特定的DNET转发Router-Busy-To-Network信息

目的：验证IUT是否正确转发了由于通信拥塞而暂时无法访问特定DNET的信息。

测试步骤：

1. TRANSMIT PORT B,

DESTINATION = LOCAL BROADCAST,

SOURCE = R2-3,

Router-Busy-To-Network,

Network Numbers = 6

1. RECEIVE PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = IUT,

Router-Busy-To-Network,

Network Numbers = 6

1. 向所有DNET发送Router-Busy-To-Network信息

目的：为了验证IUT是否正确转发了由于流量拥塞，暂时无法通过特定路由器访问的所有DNET的信息。

测试步骤：

1. TRANSMIT PORT B,

DESTINATION = LOCAL BROADCAST,

SOURCE = R2-3,

Router-Busy-To-Network

1. RECEIVE PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = IUT,

Router-Busy-To-Network,

Network Numbers = 3, 6 | 6, 3

1. 接收忙碌路由器的消息

目的：验证IUT拒绝向忙碌路由器发送消息。

测试步骤：

1. TRANSMIT PORT B,

DESTINATION = LOCAL BROADCAST,

SOURCE = R2-3,

Router-Busy-To-Network,

Network Numbers = 3

1. RECEIVE PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = IUT,

Router-Busy-To-Network,

Network Numbers = 3

1. TRANSMIT PORT A,

DA = IUT,

SOURCE = D1A,

DNET = 3,

DADR = D3D,

Hop Count = 255,

ReadProperty-Request,

'Object Identifier' = (any BACnet standard object),

'Property Identifier' = (any required property of the specified object)

1. RECEIVE PORT A,

DESTINATION = D1A,

SOURCE = IUT,

Reject-Message-To-Network,

Reject Reason = 2 (router busy),

DNET = 3

1. 超时

目的：在繁忙计时器到期后，验证IUT是否恢复DNET的可用性状态。

测试步骤：

1. TRANSMIT PORT B,

DESTINATION = LOCAL BROADCAST,

SOURCE = R2-3,

Router-Busy-To-Network,

Network Numbers = 3

1. RECEIVE PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = IUT,

Router-Busy-To-Network,

Network Numbers = 3

1. WAIT (30 seconds)
2. TRANSMIT PORT A,

DA = IUT,

SOURCE = D1A,

DNET = 3,

DADR = D3D,

Hop Count = 255,

ReadProperty-Request,

'Object Identifier' = (any BACnet standard object),

'Property Identifier' = (any required property of the specified object)

1. RECEIVE PORT B,

DA = R2-3,

SOURCE = IUT,

DNET = 3,

DADR = D3D,

Hop Count = (any integer x: 0 < x < 255),

ReadProperty-Request,

'Object Identifier' = (the object identifier used in step 4),

'Property Identifier' = (the property identifier used in step 4)

1. 执行Router-Available-To-Network

BACnet参考条款：6.6.3.7

1. 恢复指定的DNET

目的：验证当接收到传送特定目标网络的Router-Available-To-Network消息时，IUT会更新其网络可用性信息。

测试步骤：

1. TRANSMIT PORT B,

DESTINATION = LOCAL BROADCAST,

SOURCE = R2-3,

Router-Busy-To-Network

1. RECEIVE PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = IUT,

Router-Busy-To-Network,

Network Numbers = 3, 6 | 6, 3

1. TRANSMIT PORT B,

DESTINATION = LOCAL BROADCAST,

SOURCE = R2-3,

Router-Available-To-Network,

Network Numbers = 3

1. RECEIVE PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = IUT,

Router-Available-To-Network,

Network Numbers = 3

1. TRANSMIT PORT A,

DESTINATION = IUT,

SOURCE = D1A,

DNET = 3,

DADR = D3D,

Hop Count = 255,

ReadProperty-Request,

'Object Identifier' = (any BACnet standard object),

'Property Identifier' = (any required property of the specified object)

1. RECEIVE PORT B,

DESTINATION = R2-3,

SOURCE = IUT,

DNET = 3,

DADR = D3D,

Hop Count = (any integer x: 0 < x < 255),

ReadProperty-Request,

'Object Identifier' = (the object identifier used in step 5),

'Property Identifier' = (the property identifier used in step 5)

1. TRANSMIT PORT A,

DESTINATION = R2-3,

SOURCE = IUT,

DNET = 3,

DADR = D3D,

Hop Count = (any integer x: 0 < x < 255),

ReadProperty-Request,

'Object Identifier' = (the object identifier used in step 5),

'Property Identifier' = (the property identifier used in step 5)

1. RECEIVE PORT A,

DESTINATION = D1A,

SOURCE = IUT,

Reject-Message-To-Network,

Reject Reason = 2 (router busy),

DNET = 6

1. 恢复所有的DNET

目的：验证当收到一条未指定目标网络的Router-Available-To-Network消息时，IUT更新其网络可用信息。

测试步骤：

1. TRANSMIT PORT B,

DESTINATION = LOCAL BROADCAST,

SOURCE = R2-3,

Router-Busy-To-Network

1. RECEIVE PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = IUT,

Router-Busy-To-Network,

Network Numbers = 3, 6 | 6, 3

1. TRANSMIT PORT B,

DESTINATION = LOCAL BROADCAST,

SOURCE = R2-3,

Router-Available-To-Network

1. RECEIVE PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = IUT,

Router-Available-To-Network,

Network Numbers = 3, 6 | 6, 3

1. TRANSMIT PORT A,

DA = IUT,

SOURCE = D1A,

DNET = 3,

DADR = D3D,

Hop Count = 255,

ReadProperty-Request,

'Object Identifier' = (any BACnet standard object),

'Property Identifier' = (any required property of the specified object)

1. RECEIVE PORT B,

DA = R2-3,

SOURCE = IUT,

DNET = 3,

DADR = D3D,

Hop Count = (any integer x: 0 < x < 255),

ReadProperty-Request,

'Object Identifier' = (the object identifier used in step 5),

'Property Identifier' = (the property identifier used in step 5)

1. TRANSMIT PORT A,

DA = IUT,

SOURCE = D1A,

DNET = 6,

DADR = (any valid device address),

Hop Count = 255,

ReadProperty-Request,

'Object Identifier' = (any BACnet standard object),

'Property Identifier' = (any required property of the specified object)

1. RECEIVE PORT B,

DA = R2-3,

SOURCE = IUT,

DNET = 6,

DADR = (the address used in step 6),

Hop Count = (any integer x: 0 < x < 255),

ReadProperty-Request,

'Object Identifier' = (the object identifier used in step 7),

'Property Identifier' = (the property identifier used in step 7)

1. 执行初始化路由表
2. 查询路由表

所需测试：11.2.2.1, 11.2.2.2.5

BACnet参考条款：6.4.7，6.4.8，6.6.1，6.6.3.8和6.6.3.9

目的：验证IUT可正确响应端口数包括0的Initialize-Routing-Table消息。

测试步骤：

1. TRANSMIT PORT A,

DESTINATION = IUT,

SOURCE = D1A,

Initialize-Routing-Table,

Number of Ports = 0

1. RECEIVE PORT A,

DESTINATION = D1A,

SOURCE = IUT,

Initialize-Routing-Table-Ack,

Number of Ports = 5,

Connected DNET = 1,

Port ID = 1,

Port Info = (any valid port information),

Connected DNET = 2,

Port ID = 2,

Port Info = (any valid port information),

Connected DNET = 3

Port ID = 2,

Port Info = (any valid port information),

Connected DNET = 5,

Port ID = 1,

Port Info = (any valid port information),

Connected DNET = 6,

Port ID = 2,

Port Info = (any valid port information)

注意事项：DNET表条目可以是任何次序。

1. 在路由表增加条目

所需测试：11.2.2.1，11.2.2.2.5

BACnet参考条款：6.4.7,6.4.8,6.6.1,6.6.3.8,和6.6.3.9.

目的：验证当收到一条端口数非0值的Initialize-Routing-Table消息时，IUT正确应答并更新其路由表。

测试步骤：

1. TRANSMIT PORT A,

DESTINATION=IUT,

SOURCE=D1A,

Initialize-Routing-Table,

Number of Ports= 1,

Connected DNET= 1234,

PortID = 1,

Port Info Length = 0

1. RECEIVE PORT A,

DESTINATION=D1A,

SOURCE=IUT,

Initialize-Routing-Table-Ack,

1. TRANSMIT PORT A,

DESTINATION=D1A,

SOURCE=IUT,

Initialize-Routing-Table,

Number of Ports= 0,

1. RECEIVE PORT A,

DESTINATION=D1A,

SOURCE=IUT,

Initialize-Routing-Table-Ack,

Number of Ports = 6,

Connected DNET = 1,

Port ID = 1,

Port Info = (any valid port information),

Connected DNET = 2,

Port ID = 2,

Port Info = (any valid port information),

Connected DNET = 3,

Port ID = 2,

Port Info = (any valid port information),

Connected DNET = 5,

Port ID = 1,

Port Info = (any valid port information),

Connected DNET = 6,

Port ID = 2,

Port Info = (any valid port information),

Connected DNET = 1234,

Port ID = 1,

Port Info = (any valid port information)

注意事项：DNET表条目可以是任何次序。

1. 删除路由表中条目

所需测试：11.2.2.1,11.2.2.2.5

BACnet参考条款：6.4.7,6.4.8,6.6.1,6.6.3.8,和6.6.3.9.

目的：验证当收到一条Port ID为0的Initialize-Routing-Table消息时，IUT能正确响应。

测试步骤：

1. TRANSMIT PORT A,

DESTINATION = IUT,

SOURCE = D1A,

Initialize-Routing-Table,

Number of Ports = 1,

Connected DNET = 1234,

Port ID = 1,

Port Info Length = 0

1. RECEIVE PORT A,

DESTINATION = D1A,

SOURCE = IUT,

Initialize-Routing-Table-Ack

1. TRANSMIT PORT A,

DESTINATION = IUT,

SOURCE = D1A,

Initialize-Routing-Table,

Number of Ports = 0,

1. RECEIVE PORT A,

DESTINATION = D1A,

SOURCE = IUT,

Initialize-Routing-Table-Ack,

Number of ports = 4,

Connected DNET = 1,

Port ID = 1,

Port Info = (any valid port information),

Connected DNET = 2,

Port ID = 2,

Port Info = (any valid port information),

Connected DNET = 3,

Port ID = 2,

Port Info = (any valid port information),

Connected DNET = 5,

Port ID = 1,

Port Info = (any valid port information)

注意事项：DNET表条目可以是任何次序。

1. Reject-Message-To-Network

本节测试了网络层拒绝消息的某些可能情况。11.2.2.4涵盖了由于路由器繁忙而导致的拒绝的情况。

BACnet参考条款：6.4.4和6.6.3.5

1. 未知网络

目的：为了验证IUT拒绝向未知且无法访问的DNET上的设备发送消息。

测试步骤：

1. TRANSMIT PORT A,

DA = IUT,

SOURCE = D1A,

DNET = 9,

DADR = (any valid MAC address),

Hop Count = 255,

BACnet-Confirmed-Request-PDU,

'Service Choice' = ReadProperty-Request,

'Object Identifier' = (any object identifier),

'Property Identifier' = (any property of the specified object)

1. RECEIVE PORT B,

DESTINATION = LOCAL BROADCAST,

SOURCE = IUT,

Who-Is-Router-To-Network,

Network Number = 9

1. RECEIVE PORT A,

DESTINATION = D1A,

SOURCE = IUT,

Reject-Message-To-Network,

Reject Reason = 1, (unknown destination network)

DNET = 9

1. 未知网络层信息类型

目的：为了验证IUT将拒绝定向到IUT的包含在预留给ASHRAE的消息类型范围内的未知消息类型的网络层消息。

测试步骤：

1. TRANSMIT PORT A,

DESTINATION = IUT,

SOURCE = D1A,

Message Type = (any value in the range reserved for use by ASHRAE that is undefined in the

protocol revision claimed by the device)

1. RECEIVE PORT A,

DESTINATION = TD,

SOURCE = IUT,

Reject-Message-To-Network,

Reject Reason = 3 (unknown network layer message type),

DNET = (any value)

1. 其他未知的网络层消息类型

目的：此测试用例验证当将消息发送到其他地方时，IUT不会拒绝具有未知消息类型的网络层消息。如果IUT没有Protocol\_Revision属性或其值小于4，则不进行此测试。

BACnet参考条款：6.6.3.5

测试步骤：

1. TRANSMIT PORT A, DA = IUT, SA = D1A,

DNET = 2,

DADR = D2C,

Hop Count = 255,

Message Type = (any value in the range reserved for use by ASHRAE)

1. RECEIVE Port B, DA = D2C, SA = IUT,

SNET = 1,

SADR = D1A,

Message Type = (value from step 1)

1. TRANSMIT PORT A, DA = LOCAL BROADCAST, SA = D1A,

DNET = GLOBAL BROADCAST,

DLEN = 0,

Hop Count = 255,

Message Type = (any value in the range reserved for use by ASHRAE)

1. RECEIVE PORT B, DA = LOCAL BROADCAST, SA = IUT,

DNET = GLOBAL BROADCAST,

DLEN = 0,

SNET = 1,

SADR = D1A,

Hop Count = (any value greater than 1 and less than 255),

Message Type = (value from step 3)

1. TRANSMIT PORT A, DA = IUT, SA = D1A,

DNET = 2,

DADR = D2C,

Hop Count = 255,

Message Type = (any value in the range available for vendor proprietary messages),

Vendor ID = (any value, when paired with Message Type, that is not supported by the IUT)

1. RECEIVE Port B, DA = D2C, SA = IUT,

SNET = 1,

SADR = D1A,

Message Type = (value from step 1)

1. TRANSMIT PORT A, DA = LOCAL BROADCAST, SA = D1A,

DNET = GLOBAL BROADCAST,

DLEN = 0,

Hop Count = 255,

Message Type = (any value in the range available for vendor proprietary messages),

Vendor ID = (any value, when paired with Message Type, that is not supported by the IUT)

1. RECEIVE PORT B, DA = LOCAL BROADCAST, SA = IUT,

DNET = GLOBAL BROADCAST,

DLEN = 0,

SNET = 1,

SADR = D1A,

Hop Count = (any value greater than 1 and less than 255),

Message Type = (value from step 7)

1. 单播APDU的路由

此测试验证IUT是否正确路由使用单播目标地址的消息

BACnet参考条款：6.2.2和6.5

1. 忽略本地消息流

目的：验证IUT会忽略本地单播消息流量

测试步骤：

1. TRANSMIT PORT A,

DESTINATION = D1B,

SOURCE = D1A,

BACnet-Confirmed-Request-PDU,

'Service Choice' = ReadProperty-Request,

'Object Identifier' = (any object identifier),

'Property Identifier' = (any property of the specified object)

1. CHECK (verify that this message is not forwarded to Network 2)
2. 从一个本地设备到另一个本地设备的路由消息

目的：验证IUT能路由一条从网络1上的一个本地设备到网络2上的一个设备的一条单播消息。

测试步骤：

1. TRANSMIT PORT A,

DA = IUT,

SA = D1A,

DNET = 2,

DADR = D2C,

Hop Count = 255,

BACnet-Confirmed-Request-PDU,

'Service Choice' = ReadProperty-Request,

'Object Identifier' = (any object identifier),

'Property Identifier' = (any property of the specified object)

1. RECEIVE PORT B,

DA = D2C,

SA = IUT,

SNET = 1,

SADR = D1A,

BACnet-Confirmed-Request-PDU,

'Service Choice' = ReadProperty-Request,

'Object Identifier' = (the object identifier used in step 1),

'Property Identifier' = (the property identifier used in step 1)

1. TRANSMIT PORT B,

DA = IUT,

SA = D2C,

DNET = 1,

DADR = D1A,

Hop Count = 255,

BACnet-Confirmed-Request-PDU,

'Service Choice' = ReadProperty-Request,

'Object Identifier' = (any object identifier),

'Property Identifier' = (any property of the specified object)

1. RECEIVE PORT A,

DA = D1A,

SA = IUT,

SNET = 2,

SADR = D2C,

BACnet-Confirmed-Request-PDU,

'Service Choice' = ReadProperty-Request,

'Object Identifier' = (the object identifier used in step 1),

'Property Identifier' = (the property identifier used in step 1)

1. 将消息从一个本地设备路由到一个路由器

目的：验证IUT是否可以将单播消息从网络1上的本地设备路由到目标网络路径上的路由器。

测试步骤：

1. TRANSMIT PORT A,

DA = IUT,

SOURCE = D1A,

DNET = 3,

DADR = D3D,

Hop Count = 255,

BACnet-Confirmed-Request-PDU,

'Service Choice' = ReadProperty-Request,

'Object Identifier’= (any object identifier),

'Property Identifier’= (any property of the specified object)

1. RECEIVE PORT B,

DA = R2-3,

SA = IUT,

DNET = 3,

DADR = D3D,

SNET = 1,

SADR = D1A,

Hop Count= (any integer x: 0 < x < 255)

BACnet-Confirmed-Request-PDU,

'Service-Choice'= ReadProperty-Request,

'Object Identifier'= (the object identifier used in step 1),

'Property Identifier'= (the property identifier used in step 1)

1. 消息从一个路由器路由到另一个路由器

目的：验证IUT是否可以将单播消息从远程网络上的设备路由到目标网络路径上的路由器。

测试步骤：

1. TRANSMIT PORT A,

DA = IUT,

SA = R1-5,

DNET = 3,

DADR = D3D,

SNET = 5,

SADR = D5F,

Hop Count = 254,

BACnet-Confirmed-Request-PDU,

'Service Choice' = ReadProperty-Request,

'Object Identifier'= (any object identifier),

'Property Identifier'= (any property of the specified object)

1. RECEIVE PORT B,

DA = R2-3,

SA = IUT,

DNET = 3,

DADR = D3D,

SNET = 5,

SADR = D5F,

Hop Count = (and integer x: 0 < x < 254),

BACnet-Confirmed-Request-PDU,

'Service Choice' = ReadProperty-Request,

'Object Identifier'= (the object identifier used in step 1),

'Property Identifier'= (the property identifier used in step 1)

1. 从一个路由器到另一个本地设备的路由信息

目标: 验证IUT是否可以将单播消息从一个端路由器路由到一个本地网络上的目标设备。

测试步骤:

1. TRANSMIT PORT A,

DA = IUT,

SA = R1-5,

DNET = 2,

DADR = D2C,

SNET = 5,

SADR = D5F,

Hop Count = 254,

BACnet-Confirmed-Request-PDU,

'Service Choice' = ReadProperty-Request,

'Object Identifier' = (any object identifier),

'Property Identifier' = (any property of the specified object)

1. RECEIVE PORT B,

DA = D2C,

SA = IUT,

SNET = 5,

SADR = D5F,

BACnet-Confirmed-Request-PDU,

'Service Choice' = ReadProperty-Request,

'Object Identifier' = (the object identifier used in step 1),

'Property Identifier' = (the property identifier used in step 1)

1. 尝试定位网络下游路由器

本节测试IUT的搜索未知目标网络的能力。

配置需求：IUT应被配置为仅仅知道直连的网络的信息。

1. 尝试定位路由器失败

目的：证明IUT将尝试对未知网络的路由器定位。如果找不到这样的路由器，IUT将向源设备发送Reject-Message-To-Network消息。

测试步骤：

1. TRANSMIT PORT A,

DA = IUT,

SA = R1-5,

DNET = 3,

DADR = D3D,

SNET = 5,

SADR = D5F,

Hop Count = 254,

BACnet-Confirmed-Request-PDU,

'Service Choice' = ReadProperty-Request,

'Object Identifier'= (any object identifier),

'Property Identifier'= (any property of the specified object)

1. RECEIVE PORT B,

DESTINATION = LOCAL BROADCAST,

SOURCE = IUT,

Who-Is-Router-To-Network,

Network Number = 3

1. RECEIVE PORT A,

DA = R1-5,

SOURCE = IUT,

DNET = 5,

DADR = D5F,

Hop Count = 255,

Reject-Message-To-Network,

Reject Reason = 1 (unknown destination network),

DNET = 3

1. 尝试定位路由器成功

目的：验证IUT尝试对未知网络的路由器定位。如果成功找到这样的路由器，IUT将向路径上的下一个路由器发送消息。

测试步骤：

1. TRANSMIT PORT A,

DA = IUT,

SA = R1-5,

DNET = 3,

DADR = D3D,

SNET = 5,

SADR = D5F,

Hop Count = 254,

BACnet-Confirmed-Request-PDU,

'Service Choice' = ReadProperty-Request,

'Object Identifier'= (any object identifier),

'Property Identifier'= (any property of the specified object)

1. RECEIVE PORT B,

DESTINATION = LOCAL BROADCAST,

SOURCE = IUT,

Who-Is-Router-To-Network,

Network Number = 3

1. TRANSMIT PORT B,

DESTINATION = LOCAL BROADCAST,

SOURCE = R2-3,

I-Am-Router-To-Network,

Network Numbers = 3

1. RECEIVE PORT B,

SA = R2-3,

SA = IUT,

DNET = 3,

DADR = D3D,

SNET = 5,

SADR = D5F,

Hop Count= (any integer x: 0 < x < 255)

BACnet-Confirmed-Request-PDU,

'Service-Choice'= ReadProperty-Request,

'Object Identifier'= (the object identifier used in step 1),

'Property Identifier'= (the property identifier used in step 1)

1. 广播APDU的路由过程

本条款测试验证IUT是否正确路由使用广播目标地址的消息。

BACnet参考条款：6.2.2和6.5

1. 忽略本地广播信息流

目的: 验证IUT将忽略不包含远程网络寻址的本地广播消息流。

测试步骤：

1. TRANSMIT PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = D1A,

BACnet-Unconfirmed-Request-PDU,

'Service Choice' = Who-Is

1. CHECK (verify that this message is not forwarded to Network 2)
2. 从本地设备进行全局广播

目的：验证IUT是否正确转发源自本地网络的全局广播消息。

测试步骤:

1. TRANSMIT PORT A,

DA = LOCAL BROADCAST,

SOURCE = D1A,

DNET = GLOBAL BROADCAST,

DLEN = 0,

Hop Count = 255,

BACnet-Unconfirmed-Request-PDU,

'Service Choice' = Who-Is

1. RECEIVE PORT B,

DA = LOCAL BROADCAST,

SA = IUT,

DNET = GLOBAL BROADCAST,

DLEN = 0,

SNET = 1,

SADR = D1A,

Hop Count = (any integer x: 0 < x < 255),

BACnet-Unconfirmed-Request-PDU,

'Service Choice' = Who-Is

注意事项：步骤2中描述的消息应从路由器的除了端口外的所有端口传输。

1. 来自远程设备的的全局广播

目的：验证IUT能否正确地转发来自远程网络的全局广播。

测试步骤：

1. TRANSMIT PORT A,

DA = LOCAL BROADCAST,

SA = R1-5,

DNET = GLOBAL BROADCAST,

DLEN = 0,

SNET = 5,

SADR = D5F,

Hop Count = 254,

BACnet-Unconfirmed-Request-PDU,

'Service Choice' = Who-Is

'I-Am Device Identifier' = D5F,

1. RECEIVE PORT B,

DA = LOCAL BROADCAST,

SA = IUT,

DNET = GLOBAL BROADCAST,

DLEN = 0,

SNET = 5,

SADR = D5F,

Hop Count = (any value x: 0 < x < 254),

BACnet-Unconfirmed-Request-PDU,

'Service Choice' = Who-Is

注意事项：步骤2中描述的消息应从路由器的所有端口（端口1除外）传输。

1. 从一个本地设备到一个直连网络的远程广播

目的：验证IUT能否正确转发来自本地网络的远程广播消息，并将其定向到另一个本地网络。

测试步骤：

1. TRANSMIT PORT B,

DA = LOCAL BROADCAST,

SA = D2C,

DNET = 1,

DLEN = 0,

Hop Count = 255,

BACnet-Unconfirmed-Request-PDU,

'Service Choice' = Who-Is

1. RECEIVE PORT A,

DA = LOCAL BROADCAST,

SA = IUT,

SNET = 2,

SADR = D2C,

BACnet-Unconfirmed-Request-PDU,

'Service Choice' = Who-Is

1. 从一个本地设备到一个非直接连接网络的远程广播

目的：验证IUT能否正确转发来自本地网络的远程广播消息，并将其定向到非直接连接的远程网络。

测试步骤：

1. TRANSMIT PORT B,

DA = LOCAL BROADCAST,

SOURCE = D2C,

DNET = 5,

DLEN = 0,

Hop Count = 255,

BACnet-Unconfirmed-Request-PDU,

'Service Choice' = Who-Is

1. RECEIVE PORT A,

DA = R1-5,

SA = IUT,

DNET= 5,

DLEN = 0,

SNET = 2,

SADR = D2C,

Hop Count = (any integer x: 0 < x < 255),

BACnet-Unconfirmed-Request-PDU,

'Service Choice' = Who-Is

1. 从一个远程设备到一个直接连接网络的远程广播

目的：验证IUT是否能正确转发来自远程网络的远程广播消息，并将其定向到直接连接的网络。

测试步骤：

1. TRANSMIT PORT B,

DA = IUT,

SA = R2-3,

DNET = 1,

DLEN = 0,

SNET = 3,

SADR = D3D,

Hop Count = 254,

BACnet-Unconfirmed-Request-PDU,

'Service Choice' = Who-Is

1. RECEIVE PORT A,

DA = LOCAL BROADCAST,

SA = IUT,

SNET = 3,

SADR = D3D,

BACnet-Unconfirmed-Request-PDU,

'Service Choice' = Who-Is

1. 从一个远程设备到一个远程网络的远程广播

目的：验证IUT是否能正确转发来自远程网络的远程广播消息，并将其定向到远程网络。

测试步骤：

1. TRANSMIT PORT A,

DA = IUT,

SA = R1-5,

DNET = 3,

DLEN = 0,

SNET = 5,

SADR = D5F,

Hop Count = 254,

BACnet-Unconfirmed-Request-PDU,

'Service Choice' = Who-Is

1. RECEIVE PORT B,

DA = R2-3,

SA = IUT,

DNET = 3,

DLEN = 0,

SNET = 5,

SADR = D5F,

Hop Count = (any integer x,: 0 < x < 254),

BACnet-Unconfirmed-Request-PDU,

'Service Choice' = Who-Is

1. 应忽略的远程广播

目的：验证IUT忽略用于可通过接收消息的同一端口访问的远程网络的广播消息。

测试步骤：

1. TRANSMIT PORT B,

DA = LOCAL BROADCAST,

SA = D2C,

DNET = 3,

DLEN = 0,

Hop Count = 255,

'Service Choice' = Who-Is

1. CHECK (verify that the IUT does not forwarded this message)
2. 跳数保护

BACnet参考条款：6.2.3,6.5.4。

目的：验证如果跳数为零，IUT将丢弃一条消息。

1. TRANSMIT PORT A,

DA = IUT,

SA = R1-5,

DNET = 3,

DLEN = 0,

SNET = 5,

SADR = D5F,

Hop Count = 1,

BACnet-Unconfirmed-Request-PDU,

'Service Choice' = Who-Is

1. CHECK (verify that the IUT does not forward this message)
2. TRANSMIT PORT A,

DA = IUT,

SA = R1-5,

DNET = 3,

DLEN = 0,

SNET = 5,

SADR = D5F,

Hop Count = 0,

BACnet-Unconfirmed-Request-PDU,

'Service Choice' = Who-Is

1. CHECK (verify that the IUT does not forward this message)
2. 网络层优先级

BACnet参考条款: 6.1, 6.2.2, 6.5.4

目的：验证IUT能否处理和转发所有网络优先级的消息。

测试步骤：

1. TRANSMIT PORT A,

DA = IUT,

SA = D1A,

Priority = B'00',

DNET = 2,

DADR = D2C,

Hop Count = 255,

BACnet-Confirmed-Request-PDU,

'Service Choice' = ReadProperty-Request,

'Object Identifier' = (any object identifier),

'Property Identifier' = (any property of the specified object)

1. RECEIVE PORT B,

SA = D2C,

SA = IUT,

Priority = B'00',

SNET = 1,

SDR = D1A,

BACnet-Confirmed-Request-PDU,

'Service Choice' = ReadProperty-Request,

'Object Identifier' = (the object identifier used in step 1),

'Property Identifier' = (the property identifier used in step 1)

1. TRANSMIT PORT A,

DA = IUT,

SA = D1A,

Priority = B'01',

DNET = 2,

DADR = D2C,

Hop Count = 255,

BACnet-Confirmed-Request-PDU,

'Service Choice' = ReadProperty-Request,

'Object Identifier' = (any object identifier),

'Property Identifier' = (any property of the specified object)

1. RECEIVE PORT B,

DA = D2C,

SA = IUT,

Priority = B'01',

SNET = 1,

SDR = D1A,

BACnet-Confirmed-Request-PDU,

Service Choice' = ReadProperty-Request,

'Object Identifier' = (the object identifier used in step 3),

'Property Identifier' = (the property identifier used in step 3)

1. TRANSMIT PORT A,

DA = IUT,

SA = D1A,

Priority = B'10',

DNET = 2,

DADR = D2C,

Hop Count = 255,

BACnet-Confirmed-Request-PDU,

'Service Choice' = ReadProperty-Request,

'Object Identifier' = (any object identifier),

'Property Identifier' = (any property of the specified object)

1. RECEIVE PORT B,

DA = D2C,

SA = IUT,

Priority = B'10',

SNET = 1,

SDR = D1A,

BACnet-Confirmed-Request-PDU,

'Service Choice' = ReadProperty-Request,

'Object Identifier' = (the object identifier used in step 5),

'Property Identifier' = (the property identifier used in step 5)

1. TRANSMIT PORT A,

DA = IUT,

SA = D1A,

Priority = B'11',

DNET = 2,

DADR = D2C,

Hop Count = 255,

BACnet-Confirmed-Request-PDU,

'Service Choice' = ReadProperty-Request,

'Object Identifier' = (any object identifier),

'Property Identifier' = (any property of the specified object)

1. RECEIVE PORT B,

DA = D2C,

SA = IUT,

Priority = B'11',

SNET = 1,

SDR = D1A,

BACnet-Confirmed-Request-PDU,

'Service Choice' = ReadProperty-Request,

'Object Identifier' = (the object identifier used in step 7),

'Property Identifier' = (the property identifier used in step 7)

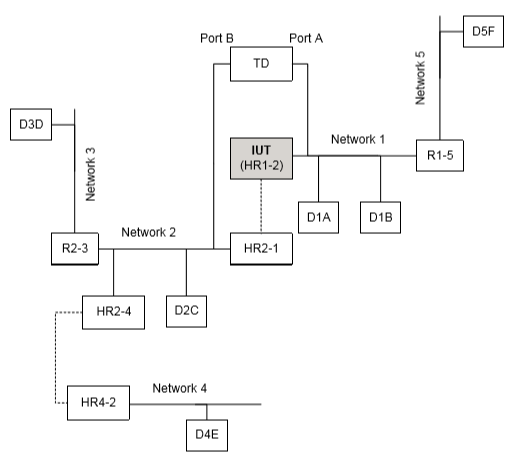
* 1. 半路由器的功能测试

本条定义了演示BACnet半路由器功能所需的测试。假定半路由器有2个端口，1口直接连到网络1，2口是一个通向网络2的PTP连接。用于这些测试的互联网的逻辑配置如图11-2所示。本条中的测试说明假设TD可以同时连接到网络1和网络2，并模拟所有其他设备，包括端半路由器HR2-1。端口A连接网络1，端口B连接网络2。一个可接受的替代方案是用实际设备构建一个网络，如图所示。逻辑网络3和5应采用不同的局域网技术，这两种技术不同于网络1和网络2，以确保路由器能够支持具有不同大小MAC地址的远程网络。除非使用了PTP连接，否则本节测试不涉及PTP功能。PTP功能测试的说明在13.2。

所需测试：PTP状态机测试，13.2

配置需求： IUT配置路由表为网络1直接连接到端口1，PTP连接可通过端口2建立到网络2，但尚未建立。路由表不应包含其他项。

半路由器功能测试应按本标准规定的次序。在某些情况下，成功完成测试用例要求IUT从上一个测试的最后的状态开始进行测试。



**图11-2** 用于半路由器功能测试的网络逻辑配置

网络上的逻辑设备说明如下：

IUT: 待测设备，连接到网络1的半路由器，它可以使PTP连接到网络2的半路由器。

D1A: 网络1上的设备

D1B: 网络1上的设备

D2C: 网络2上的设备

D3D: 网络3上的设备

D4E: 网络4上的设备

D5F: 网络5上的设备

R1-5: 连接网络1与5的路由器

R2-3: 连接网络2与3的路由器

HR2-1: 连接到网络2的半路由器，它可以使PTP连接到网络1的半路由器。

HR2-4: 连接到网络2的半路由器，它可以使PTP连接到网络4的半路由器.

1. 执行Who-Is-Router-To-Network

本条定义的测试验证IUT在PTP连接未建立之前能否正确响应Who-Is-Router-To-Network消息。

BACnet参考条款：6.6.3.4

1. 未指定网络号

目的：验证IUT当收到未指定网络号的Who-Is-Router-To-Network消息时不发出任何消息。

测试步骤：

1. TRANSMIT PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = D1A,

Who-Is-Router-To-Network

2. CHECK (verify that the IUT does not respond)

1. 指定了可以通过PTP连接到达的网络号

目的：验证IUT当收到指定了能够通过PTP连接而到达的Who-Is-Router-To-Network消息时会发出I-Could-Be-Router-To-Network消息。

测试步骤：

1. TRANSMIT PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = D1A,

Who-Is-Router-To-Network,

Network Number = 2

2. RECEIVE PORT A,

DESTINATION = D1A,

SOURCE = IUT,

I-Could-Be-Router-To-Network,

Network Number = 2,

Performance Index = (any valid performance index)

1. 若无法连接发出拒绝消息

BACnet 参考条款：6.7.4.1.1

目的：要求IUT将消息路由到未建立连接的网络，验证IUT将发送Reject-Message-To-Network消息。

测试步骤:

1. TRANSMIT PORT A,

DESTINATION = IUT,

SOURCE = D1A,

DNET = 2,

DADR = D2C,

Hop Count = 255,

ReadProperty-Request,

'Object Identifier' = (any BACnet object),

'Property Identifier' = (any property of the specified object)

1. RECEIVE PORT A,

DESTINATION = D1A,

SOURCE= IUT,

Reject-Message-To-Network,

Reject-Reason= 1 (unknown destination network),

DNET= 2

1. 启动半路由器连接建立过程

BACnet参考条款：6.7.1

目的：在启动与端半路由器的连接时，要验证IUT是否遵循正确的程序。

测试步骤：

1. TRANSMIT PORT A,

DESTINATION = IUT,

SOURCE = D1A,

Establish-Connection-To-Network,

DNET = 2,

Termination Time Value = 60

1. BEFORE (60 seconds) RECEIVE PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = IUT,

I-Am-Router-To-Network,

Network Numbers = 2

1. TRANSMIT PORT A,

DESTINATION = IUT,

SOURCE = D1A,

DNET = 2,

DADR = D2C,

Hop Count = 255,

ReadProperty-Request,

'Object Identifier' = (any object identifier),

'Property Identifier' = (any property of the specified object)

1. RECEIVE PORT B,

DESTINATION = D2C,

SOURCE = IUT,

SNET = 1,

SADR = D1A,

Hop Count = (any integer x: 0 < x < 255),

ReadProperty-Request,

'Object Identifier' = (the object identifier used in step 3),

'Property Identifier' = (the property identifier used in step 3)

1. 有效期结束自动断开连接

所需测试：11.3.3

BACnet参考条款：6.7.1.4和6.7.2.2

目的:验证如果在Termination Time Value指定的时间内没有调用IUT来路由任何消息，则IUT将终止与另一个半路由器的连接。

测试步骤：

1. WAIT (90 seconds)
2. TRANSMIT PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = D1A,

Who-Is-Router-To-Network

1. CHECK(verify that the IUT does not respond)
2. TRANSMIT PORT A,

DESTINATION = IUT,

SOURCE = D1A,

DNET = 2,

DADR = D2C,

Hop Count = 255,

ReadProperty-Request,

'Object Identifier' = (any object identifier),

'Property Identifier' = (any property of the specified object)

1. RECEIVE PORT A,

DESTINATION = D1A,

SOURCE = IUT,

Reject-Message-To-Network,

Reject Reason = 1 (unknown destination network),

DNET = 2

1. 与半路由器建立连接的应答过程

BACnet参考条款：6.7.1.3

目的：验证IUT遵循正确的程序建立由端半路由器发起的连接。

测试步骤：

1. MAKE (the TD or a real half-router HR2-1 initiate a PTP connection to the IUT with a Termination Time Value of 0)
2. BEFORE (60 seconds) RECEIVE PORT A, DESTINATION = LOCAL BROADCAST, SOURCE = IUT,

I-Am-Router-To-Network,

Network Numbers = 2

1. TRANSMIT PORT A,

DESTINATION = IUT,

SOURCE = D1A,

DNET = 2,

DADR = D2C,

Hop Count = 255,

ReadProperty-Request,

'Object Identifier' = (any BACnet object),

'Property Identifier' = (any property of the specified object)

1. RECEIVE PORT B,

DESTINATION = D2C,

SOURCE = IUT,

SNET = 1,

SADR = D1A,

Hop Count = (any integer x: 0 < x < 255),

ReadProperty-Request,

'Object Identifier' = (the object identifier used in step 3),

'Property Identifier' = (the property identifier used in step 3)

1. 周期广播I-Am-Router-To-Network,

BACnet参考条款：6.7.4.2.2

目的：验证当IUT与另一个半路由器建立连接时，IUT将每隔五分钟广播一条I-Am-Router-To-Network消息。

测试步骤：

1. BEFORE (5 minutes) {

RECEIVE PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = IUT,

I-Am-Router-To-Network,

Network Numbers = 2

RECEIVE PORT B,

DESTINATION = LOCAL BROADCAST,

SOURCE = HR2-1,

I-Am-Router-To-Network,

Network Numbers = 1}

1. BEFORE (5 minutes) {

RECEIVE PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = IUT,

I-Am-Router-To-Network,

Network Numbers = 2

RECEIVE PORT B,

DESTINATION = LOCAL BROADCAST,

SOURCE = HR2-1,

I-Am-Router-To-Network,

Network Numbers = 1}

1. BEFORE (5 minutes) {

RECEIVE PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = IUT,

I-Am-Router-To-Network,

Network Numbers = 2

RECEIVE PORT B,

DESTINATION = LOCAL BROADCAST,

SOURCE = HR2-1,

I-Am-Router-To-Network,

Network Numbers = 1}

注意事项：I-Am-Router-To-Network消息能以任何次序接收，时间间隔大约是5分钟。

1. 断开网络连接

所需测试：11.3.5

BACnet参考条款：6.7.2.1

目的：验证IUT在接收到Disconnect-Connection-To-Network消息后，将遵循正确的过程来终止与端半路由器的连接。

测试步骤：

1. TRANSMIT PORT A,

DESTINATION = IUT,

SOURCE = D1A,

Disconnect-Connection-To-Network,

DNET = 2

1. WAIT (60 seconds)
2. TRANSMIT PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = D1A,

Who-Is-Router-To-Network

1. CHECK(verify that the IUT does not respond)
2. TRANSMIT PORT A,

DESTINATION = IUT,

SOURCE = D1A,

DNET = 2,

DADR = D2C,

Hop Count = 255,

ReadProperty-Request,

'Object Identifier' = (any BACnet object),

'Property Identifier' = (any property of the specified object)

1. RECEIVE PORT A,

DESTINATION = D1A,

SOURCE = IUT,

Reject-Message-To-Network,

Reject Reason = 1 (unknown destination network),

DNET = 2

1. 从重复的网络连接中恢复

BACnet参考条款：6.7.4.2

目的：如果收到一条包含到半路由器直接连接的网络之一的DNET的I-Am-Router-To-Network消息，验证IUT将终止与另一个半路由器的连接。

测试步骤：

1. TRANSMIT PORT A,

DESTINATION = IUT,

SOURCE = D1A,

Establish-Connection-To-Network,

DNET = 2,

Termination Time Value = 60

1. BEFORE (60 seconds) RECEIVE PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = IUT,

I-Am-Router-To-Network,

Network Numbers = 2

1. TRANSMIT PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = R1-5,

I-Am-Router-To-Network,

Network Numbers = 2,5

1. WAIT (60 seconds)
2. TRANSMIT PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = D1A,

Who-Is-Router-To-Network

1. CHECK(verify that the IUT does not respond)
2. TRANSMIT PORT A,

DESTINATION = IUT,

SOURCE = D1A,

DNET = 2,

DADR = D2C,

Hop Count = 255,

ReadProperty-Request,

'Object Identifier' = (any BACnet object),

'Property Identifier'= (any property of the specified object)

1. RECEIVE PORT A,

DESTINATION = D1A,

SOURCE = IUT,

Reject-Message-To-Network,

Reject Reason = 1 (unknown destination network),

DNET = 2

1. 正常路由功能

BACnet参考条款：6.6

目的：验证，在建立PTP连接之后，IUT与其半路由器端一起充当正常的BACnet路由器。

测试思想：与端半路由器建立了PTP连接，并进行了11.2中规定的所有测试，并考虑到两个半路由器是充当这些测试的IUT的单个路由器。

测试步骤：

1. TRANSMIT PORT A,

DESTINATION = IUT,

SOURCE = D1A,

Establish-Connection-To-Network,

DNET = 2,

Termination Time Value = 0

1. BEFORE (60 seconds) RECEIVE PORT A,

DESTINATION = LOCAL BROADCAST,

SOURCE = IUT,

I-Am-Router-To-Network,

Network Numbers = 2

1. MAKE (conduct all of the tests prescribed in 10.2)

测试注意事项：IUT应按11.2中所描述的予以响应。

* 1. B/IP PAD测试

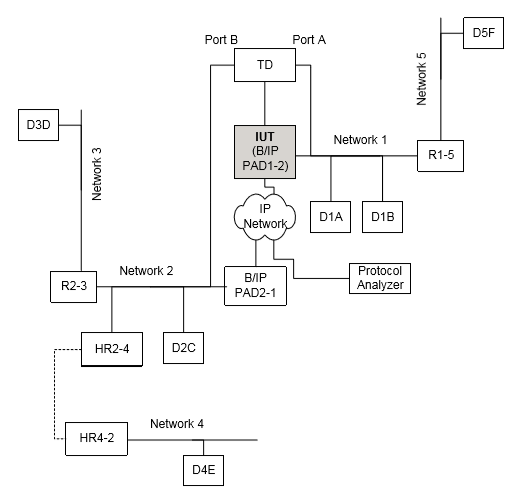
本条规定了证明BACnet/Internet协议PAD功能所需的测试，如BACnet标准附录H所述。用于这些测试的网络配置逻辑如图11-3所示。测试说明假设TD可以同时连接到网络1和网络2，并模拟所有其他设备，并监控B/IP PAD之间的IP流量。到网络1的连接称为“端口A”，到网络2的连接称为“端口B”。一个可接受的替代方案是用实际设备构建一个网络，如图所示。逻辑网络3和5应采用不同的局域网技术，这两种技术不同于网络1和网络2，以确保路由器能够支持具有不同大小MAC地址的远程网络。

配置需求：IUT应配置路由表，表明网络1直接连接到端口1，IP连接到端口2。IUT可以对两个端口使用相同的网络接口，但它们在路由表中应是不同的。测试开始前，应建立B/IP PAD2-1的IP连接。如何初始化IUT的细节是本地问题。可能需要使用域名服务器，如图11-3所示。

BACnet参考条款：附录H

测试步骤：应进行11.2中规定的所有试验

注意事项：传递结果与11.2中定义的相同，另外还应监控端B/IP路由器之间的消息，以确保它们是源端口和目标端口设置为X'BAC0'的UDP消息。



**图 11-3** B/IP路由器功能测试的网络配置逻辑

网络上的逻辑设备如下：

IUT: 待测设备。连接到网络1的B/IP路由器，配置为识别连接到网络2的端B/IP路由器。

D1A: 网络1上的设备

D1B: 网络1上的设备

D2C: 网络2上的设备

D3D: 网络3上的设备

D4E: 网络4上的设备

D5F: 网络5上的设备

R1-5: 连接网络1与5的路由器

R2-3: 连接网络2与3的路由器

HR2-4: 连接到网络2的半路由器，与连接到网络4的半路由器HR4-2构成PTP连接

* 1. 初始化网络层消息

本条规定了必要的测试证明能够初始化用于定位路由器、管理路由器表以及建立或终止PTP连接的BACnet网络层消息。这些测试不限于BACnet路由器，可应用于测试执行这些功能的任何BACnet设备。

所需测试：无

1. 定位路由器

BACnet参考条款：6.4.1。

1. Who-Is-Router-To-Network 通用查询

目的：验证IUT能正确发出格式化的未指定网络号的Who-Is-Router-To-Network消息

测试步骤：

1. MAKE (the IUT initiate a Who-Is-Router-To-Network query with no DNET specified)

2. RECEIVE

DESTINATION = LOCAL BROADCAST,

SOURCE = IUT,

Who-Is-Router-To-Network

1. Who-Is-Router-To-Network -Specific网络编号

目的：验证IUT是否可以正确传输格式化的指定网络号的Who-Is-Router-To-Network消息。

测试步骤：

1. MAKE (the IUT initiate a Who-Is-Router-To-Network query with a DNET specified)

2. RECEIVE

DESTINATION = LOCAL BROADCAST,

SOURCE = IUT,

Who-Is-Router-To-Network,

DNET = (any valid DNET)

1. 管理路由器表

BACnet参考条款：6.4.7

1. 查询路由表

目的：验证IUT能否正确发出不带端口映射的格式化的Initialize-Routing-Table消息。

测试步骤：

1. MAKE (the IUT initiate an Initialize-Routing-Table message with Number of Ports = 0)

2. RECEIVE

DESTINATION = TD,

SOURCE = IUT,

Initialize-Routing-Table,

Number of Ports = 0

1. 修改路由表

目的：验证验证IUT能否正确发出至少包含一条端口映射的格式化的Initialize-Routing-Table消息。

测试步骤：

1. MAKE (the IUT initiate an Initialize-Routing-Table message with Number of Ports > 0)
2. RECEIVE

DESTINATION = TD,

SOURCE = IUT,

Initialize-Routing-Table,

Number of Ports = (any integer > 0),

Connected DNET = (any valid DNET),

Port ID = (any valid port ID),

Port Info = (any valid port information)

注意事项：连接的目标网络号、端口ID和端口信息字段应等等于端口数量参数中指定的次数。

1. 初始化和终止PTP连接
2. Establish-Connection-To-Network

BACnet参考条款：6.4.9

目的：验证IUT能否正确发送Establish-Connection-To-Network消息

测试步骤：

1. MAKE (the IUT initiate an Establish-Connection-To-Network message)

2. RECEIVE

DESTINATION = TD,

SOURCE = IUT,

Establish-Connection-To-Network,

DNET = (any valid DNET),

Termination Time Value = (any integer ≥ 0)

1. 断开网络连接

BACnet参考条款：6.4.10

目的：验证IUT可以传输格式正确的Disconnect-Connection-To-network消息。

测试步骤：

1. MAKE (the IUT initiate an Disconnect-Connection-To-Network message)
2. RECEIVE

DESTINATION = TD,

SOURCE = IUT,

Disconnect-Connection-To-Network,

DNET = (any valid DNET)

* 1. 非路由器功能测试

本节定义了证明非路由器节点所独有的BACnet网络层功能部分的必要测试。这些测试验证节点是否能正确忽略为路由器保留的消息。如果IUT只能配置为BACnet路由器，则应省略这些测试。

测试假设IUT位于网络DNET1上，TD是网络DNET2上的路由器。值DNET3被分配一个唯一的网络号。如果IUT能发出请求，它将被配置为将这些请求发送到网络DNET2上的设备(D2A)。预计IUT会响应来自设备D2A的请求。测试假设TD模拟设备D2A。

注意事项：第6.5.1条和第6.5.3条表明，非路由器仅有两种方式传输请求（在本地网络上，目的地为远程网络），这两种方式均不包括网络层源路由信息。如果在本条款中的测试期间，IUT发出任何带有SNET/SADR字段的NPDU，则测试失败。

注意事项：第6.6和6.6.3.3节定义了BACnet路由器和为路由器保留的网络层服务。如果在本小节中的测试期间（包括在测试11.X.2中），IUT发出任何I-AM-Router-to-Network或I-Could-Be-Router-To-Network的NPDU，那么它将失败。

1. 忽视远程数据包

BACnet参考条款：6.5.2.1,6.5.4

目的：验证非路由器IUT会接收并丢弃以远程网络为目标的数据包。

测试思想：TD使用网络层头段中的DNET (不等于x’FFFF’)和DADR将广播和定向请求传输到IUT。由于IUT不是路由器，因此需要它删除请求。

测试步骤：

1. TRANSMIT

DA = BROADCAST,

SA = TD,

DNET = DNET3,

DADR = BROADCAST,

Hop Count = 255,

BACnet-Unconfirmed-Request-PDU,

'Service Choice'= who-Is

1. WAIT **Internal Processing Fail Time**
2. CHECK (that the IUT did not send an I-Am)
3. TRANSMIT

DA = IUT,

SA = TD,

DNET = DNET3,

DADR = IUT,

Hop Count = 255,

BACnet-Confirmed-Request-PDU,

'Service Choice' = ReadProperty-Request,

'Object Identifier' = (O2, any BACnet standard object in IUT),

'Property Identifier' = (P2, any required property of the specified object)

1. WAIT **Internal Processing Fail Time**
2. CHECK (that the IUT did not send a response to the ReadProperty)
3. 11.X.2 忽略 Who-Is-Router-To-Network请求

BACnet参考条款：6.6,6.6.3.2

目的：此测试验证非路由器IUT会接收并丢弃Who-Is-Router-To-Network服务。

测试思想：TD同时传输的一般查询和特定网络号查询的Who-Is-Router-To-Network服务。由于IUT不是路由器，因此需要它删除请求。

测试步骤：

1. TRANSMIT

DA = BROADCAST,

SA = TD,

Who-Is-Router-To-Network,

DNET = DNET2

1. WAIT **Internal Processing Fail Time**
2. CHECK (that the IUT did not send any packets in response to the Who-Is-Router-To-Network)
3. TRANSMIT

DA = BROADCAST,

SA = TD,

Who-Is-Router-To-Network

1. WAIT **Internal Processing Fail Time**
2. CHECK (that the IUT did not send any packets in response to the Who-Is-Router-To-Network)
3. 忽略路由器命令

BACnet参考条款：6.6,6.6.3.8,6.6.3.10,6.6.3.11

目的：此测试验证非路由器IUT会接收并丢弃网络层路由器服务。

测试思想：TD发出Initialize-Routing-Table,Establish-Connected-To-Network,和Disconnect-Connect-To-Network服务，因为IUT不是路由器，所以应当丢弃。

测试步骤：

1. TRANSMIT

DA = IUT,

SA = TD,

Initialize-Routing-Table

Number of Ports = 0

1. WAIT **Internal Processing Fail Time**
2. CHECK (that the IUT did not send any packets in response to the Initialize-Routing-Table)
3. TRANSMIT

DA = IUT,

SA = TD,

Establish-Connection-To-Network

DNET = DNET3

Termination Time Value = 0

1. WAIT **Internal Processing Fail Time**
2. CHECK (that the IUT did not send any packets in response to the Establish-Connection-To-Network)
3. TRANSMIT

DA = IUT,

SA = TD,

Disconnect-Connection-To-Network

DNET = NET3

1. WAIT **Internal Processing Fail Time**
2. CHECK(that the IUT did not send any packets in response to the Disconnect-Connection-To-Network)
   1. 路由绑定测试

此子条款定义了证明用于确定网络路由的BACnet网络层功能部分所需的测试。这些测试是通用的，应用于路由器和非路由器节点。

测试假设IUT位于网络DNET1上，TD似乎是通往DNET2的路由器。值Dnet3被分配一个唯一的网络号。如果IUT可以启动请求，它将被配置为将这些请求发送到网络DNET2上的设备（D2A）。预计IUT也将响应来自设备D2A的请求。测试假设TD模拟设备D2A。

注：第6.5.1条和第6.5.3条表明，非路由器只有两种方式来传输请求（在本地网络上并发送到远程网络），这两种方式都不包括网络层源路由信息。如果IUT被配置成一个非路由器，在此测试过程中它发出了带有SNET/SADR项的NPDU，则测试失败。

注:第6.6条和第6.6.3.3条定义了BACnet路由器和为路由器保留的网络层服务。如果IUT被配置为非路由器，且在此测试过程中发出I-Am-router-To-Network或I-Could-Be-Router-To-Network NPDUs命令，则测试失败。

1. 静态路由器绑定

所需测试：ReadProperty服务启动测试，9.15，ReadProperty服务执行测试，10.15

BACnet参考条款：6.5.3

目的：验证使用路由器的MAC地址静态配置IUT的情况下，此IUT可以向该远程网络发起请求。

测试思想：IUT将请求发送到远程网络上的设备，而不执行任何形式的动态路由器绑定。如果IUT不支持静态路由器绑定，或者IUT无法发起请求，则应省略该测试。如果IUT无法启动ReadProperty请求，则可以替换另一个服务。

测试配置：将IUT配置为使用TD的MAC地址作为网络DNET2的路由器。

测试步骤：

1. MAKE (IUT transmit a ReadProperty request to the D2A device on the remote network)

2. RECEIVE

DA = TD,

SA = IUT,

DNET = DNET2,

DADR= D2A,

Hop Count = 255,

BACnet-Confirmed-Request-PDU,

'Service Choice' = ReadProperty-Request,

'Object Identifier' = (O1, any BACnet standard object in D2A),

'Property Identifier' = (P1, any required property of the specified object)

3. TRANSMIT

DA = IUT,

SA = TD,

SNET = DNET2,

SADR = D2A,

BACnet-ComplexACK-PDU,

‘Service ACK Choice’ = ReadProperty-ACK,

'Object Identifier' = O1,

'Property Identifier' = P1,

'Property Value' = (any valid value)

1. 通过应用层服务的路由器绑定

所需测试：ReadProperty服务启动测试，9.18，ReadProperty服务执行测试，10.18，Who-Is服务启动测试，9.34

BACnet参考条款：6.5.3

目的：在IUT使用Who-Is和I-Am应用层服务发现到该远程网络的路由器的MAC地址之后，验证IUT可以发起对远程网络的请求并响应来自远程网络的请求。

测试思想：IUT广播Who-Is请求以发现设备D2A，并在相应的I-Am答复中记录介入路由器的MAC地址。TD将请求发送到远程网络上的设备，并响应来自远程网络的请求，而无需执行任何其他形式的动态路由器绑定。如果IUT不支持应用层路由器绑定，则省略此测试。如果IUT无法启动ReadProperty请求，则可以替换另一个确认服务。IUT可以使用Who-Is的DeviceInstanceRange形式。

第6.5.3节提到了通过Who-Is进行的路由器绑定，没有提及通过启动其他应用程序层服务（例如Who-Has）或通过记下传入应用程序层请求的路由器MAC地址来进行路由器绑定。因此，该测试仅允许通过Who-Is进行路由器绑定。

测试步骤：

1. MAKE (IUT transmit Who-Is to discover the device on the remote network)

2. RECEIVE

DA = BROADCAST,

SA = IUT,

DNET = GLOBAL BROADCAST,

Hop Count = 255,

BACnet-Unconfirmed-Request-PDU,

‘Service Choice’ = who-Is

| (DA = BROADCAST,

SA = IUT,

DNET = DNET2,

DADR= BROADCAST,

Hop Count = 255,

BACnet-Unconfirmed-Request-PDU,

‘Service Choice’ = who-Is )

3. TRANSMIT

DA = BROADCAST,

SA = TD,

SNET = DNET2,

SADR = D2A,

BACnet-Unconfirmed-Request-PDU,

‘Service Choice’ = I-Am,

'I Am Device Identifier' = (device object, instance number of D2A),

'Max APDU Length Accepted ' = (any valid value),

‘Segmentation Supported’ = (any valid value),

'Vendor ID ' = (any valid value)

4. MAKE (IUT transmit a ReadProperty request to the D2A device on the remote network)

5. RECEIVE

DA = TD,

SA = IUT,

DNET = DNET2,

DADR= D2A,

Hop Count = 255,

BACnet-Confirmed-Request-PDU,

'Service Choice' = ReadProperty-Request,

'Object Identifier' = (O1, any BACnet standard object in D2A),

'Property Identifier' = (P1, any required property of the specified object)

6. TRANSMIT

DA = IUT,

SA = TD,

SNET = DNET2,

SADR = D2A,

BACnet-ComplexACK-PDU,

‘Service ACK Choice’ = ReadProperty-ACK,

'Object Identifier' = O1,

'Property Identifier' = P1,

'Property Value' = (any valid value)

1. 通过Who-Is-Router-To-Network的路由器绑定

所需测试：ReadProperty服务启动测试，9.15，ReadProperty服务执行测试，10.15，定位路由器，11.5.1

BACnet参考条款：6.5.3

目的：验证在IUT使用Who-Is-Router-To-Network网络层服务查找到该远程网络的路由器的MAC地址之后， IUT可以向远程网络发起请求。

测试思想：IUT广播Who-Is-Router-To-Network请求，以发现到达所需网络的路由器。TD将请求发送到远程网络上的设备，而无需执行任何其他形式的动态路由器绑定。如果IUT不支持Who-Is-Router-To-Network路由器绑定，则应省略此测试。如果IUT无法启动ReadProperty请求，则可以替换另一个已确认的服务。 IUT可以使用Who-Is-Router-To-Network服务的常规查询或特定网络号查询形式。

注意，第6.5.3节提到了通过Who-Is-Router-To-Network进行的路由器绑定，而没有提到通过潜伏和注意未经请求的I-Am-Router-To-Network消息进行的路由器绑定。

测试步骤：

1. MAKE (IUT transmit Who-Is-Router-To-Network to discover the router to DNET2)

2. RECEIVE

DA = BROADCAST,

SA = IUT,

Who-Is-Router-To-Network,

| (DA = BROADCAST,

SA = IUT,

Who-Is-Router-To-Network,

DNET = DNET2)

3. TRANSMIT

DESTINATION = BROADCAST,

SOURCE = TD,

I-Am-Router-To-Network,

Network Numbers = DNET2

4. MAKE (IUT transmit a ReadProperty request to the D2A device on the remote network)

5. RECEIVE

DA = TD,

SA = IUT,

DNET = DNET2,

DADR= D2A,

Hop Count = 255,

BACnet-Confirmed-Request-PDU,

'Service Choice' = ReadProperty-Request,

'Object Identifier' = (O1, any BACnet standard object in D2A),

'Property Identifier' = (P1, any required property of the specified object)

6. TRANSMIT

DA = IUT,

SA = TD,

SNET = DNET2,

SADR = D2A,

BACnet-ComplexACK-PDU,

‘Service ACK Choice’ = ReadProperty-ACK,

'Object Identifier' = O1,

'Property Identifier' = P1,

'Property Value' = (any valid value)

1. 通过广播的路由器绑定

所需测试：ReadProperty服务启动测试，9.15，ReadProperty服务执行测试，10.15

BACnet参考条款：6.5.3

目的：验证当IUT使用初始广播发现到该远程网络的路由器的MAC地址时，IUT可以发起对远程网络的请求。

测试思想：IUT将请求广播到远程网络上的设备，并在回复中记录中间路由器的MAC地址。如果IUT不支持通过广播的路由器绑定，则应省略此测试。如果IUT无法启动ReadProperty请求，则可以替换另一个已确认的服务。

测试步骤：

1. MAKE (IUT transmit a ReadProperty request to the D2A device on the remote network)

2. RECEIVE

DA = BROADCAST,

SA = IUT,

DNET = DNET2,

DADR= D2A,

Hop Count = 255,

BACnet-Confirmed-Request-PDU,

'Service Choice' = ReadProperty-Request,

'Object Identifier' = (O1, any BACnet standard object in D2A),

'Property Identifier' = (P1, any required property of the specified object)

3. TRANSMIT

DA = IUT,

SA = TD,

SNET = DNET2,

SADR = D2A,

BACnet-ComplexACK-PDU,

‘Service ACK Choice’ = ReadProperty-ACK,

'Object Identifier' = O1,

'Property Identifier' = P1,

'Property Value' = (any valid value)

4. MAKE (IUT transmit a ReadProperty request to the D2A device on the remote network)

5. RECEIVE

DA = TD,

SA = IUT,

DNET = DNET2,

DADR= D2A,

Hop Count = 255,

BACnet-Confirmed-Request-PDU,

'Service Choice' = ReadProperty-Request,

'Object Identifier' = (O1, any BACnet standard object in D2A),

'Property Identifier' = (P1, any required property of the specified object)

6. TRANSMIT

DA = IUT,

SA = TD,

SNET = DNET2,

SADR = D2A,

BACnet-ComplexACK-PDU,

‘Service ACK Choice’ = ReadProperty-ACK,

'Object Identifier' = O1,

'Property Identifier' = P1,

'Property Value' = (any valid value)

1. 逻辑链路层协议测试

本条款定义了一些用来正确展示由BACnet所定义的用于ISO/IEC 8802-3（以太网）和ARCNET数据链路层中的ISO/IEC 8802-2逻辑链路控制层（LLC, 1型）的一些必要测试。这些测试都基于ISO/IEC 8802-2 (1994)，及本标准的相关参考文献。

目的服务访问点（DSAP）与源服务访问点（SSAP）的赋值定义如下：

|  |  |
| --- | --- |
| **DSAP 类型：** | **(八位)字节值：** |
| BACnet\_Individual | X'82' |
| BACnet\_Group | X'83' |
| Null\_Individual | X'00' |
| Global\_Group | X'FF' |

|  |  |
| --- | --- |
| **SSAP 类型：** | **(八位)字节值：** |
| BACnet\_Command | X'82' |
| BACnet\_Response | X'82' |
| Null\_Command | X'00' |
| Null\_Response | X'01' |

* 1. UI命令和应答

所需测试：无。

BACnet参考条款：7.1和8.1。

ISO/IEC 8802-2参考条款： 3.3, 4.1, 5.4.1.1,和5.4.1.2

目的：所有的BACnet报文都使用无编号信息（Unnumbered Information，UI）命令通过LLC Class I服务传输。本测试验证设备传输中使用了正确的DSAP和SSAP值。

测试步骤：

1. TRANSMIT ReadProperty-Request,

DSAP = BACnet\_Individual,

SSAP = BACnet\_Command,

CTL = X'03',

'Object Identifier' = (the device's Device object),

'Property Identifier' = Object\_Identifier

2. BEFORE **Acknowledgement Fail Time** RECEIVE BACnet-ComplexACK-PDU,

DSAP = BACnet\_Individual,

SSAP = BACnet\_Command,

CTL = X'03',

'Property Value' = (the device's Device object)

* 1. XID命令和应答

所需测试：无。

BACnet参考条款：7.1和8.1。

ISO/IEC 8802-2参考条款： 3.3, 4.1, 4.2.1, 5.3.1, 5.4.1.1, 5.4.1.2和6.6

目的：验证LLC对交换身份（XID）请求做出了正确的应答。

测试步骤：

1. REPEAT tSSAP = (BACnet\_Command, Null\_Command) DO {

REPEAT tDSAP = (BACnet\_Individual, BACnet\_Group,

Global\_Group, Null\_Individual) DO {

REPEAT tCTL = (X'AF', X'BF') DO {

TRANSMIT XID-Request,

DSAP = tDSAP,

SSAP = tSSAP,

CTL = tCTL,

XID-Information = X'810100'

BEFORE **Acknowledgement Fail Time** RECEIVE XID-Response,

DSAP = (BACnet\_Individual | Null\_Individual),

SSAP = (BACnet\_Response | Null\_Response),

CTL = tCTL,

XID-Information = X'810100'

}

}

}

* 1. TEST命令和应答

所需测试：无。

BACnet参考条款:7.1、8.1。

ISO/IEC 8802-2参考章节：3.3、4.1、5.3.1、5.4.1.1、5.4.1.2和6.7。

目的：验证LLC对TEST请求做出了正确的应答。

在以下测试中，任意字节的数据可以附加到TEST-Request PDU后；可以包括通过数据链路在单个PDU中传输的尽可能多的八位位组。如果IUT在TEST-Response PDU中返回数据，则它应返回与TEST-Request PDU中发送的数据相同的数据。

测试步骤：

1. REPEAT tSSAP = (BACnet\_Command, Null\_Command) DO {

REPEAT tDSAP = (BACnet\_Individual, BACnet\_Group,

Global\_Group, Null\_Individual) DO {

REPEAT tCTL = (X'E3', X'F3') DO {

TRANSMIT TEST-Request,

DSAP = tDSAP,

SSAP = tSSAP,

CTL = tCTL

BEFORE **Acknowledgement Fail Time** RECEIVE TEST-Response,

DSAP = (IF (tSSAP is Null\_Command) THEN

Null\_Individual

ELSE

BACnet\_Individual)

SSAP = ( IF (tDSAP is Null\_Individual) THEN

Null\_Response

ELSE

BACnet\_Response)

CTL = tCTL,

}

}

}

1. 数据链路层协议测试
   1. MS/TP状态机测试
2. MS/TP主测试

本节定义的测试用来验证BACnet MS/TP主设备能够正确地实现BACnet协议9.5.4节中定义的接收帧有限状态机（Receive Frame Finite State Machine）、9.5.5节中定义的发送帧(SendFrame)过程和9.5.6节定义的主节点有限状态机(Master Node Finite State Machine)。BACnet协议中的状态机图9-3和图9-4对理解该测试有帮助。

某些状态机的转换可能由不同的条件之一引发。为了区分正在执行的测试，多个条件我们用a,b,c等连续字母令牌表示。

这些测试需要在IUT支持的波特率下进行，且运行时只有IUT和TD在MS/TP LAN中工作。为了确保主设备状态机的正确，这些测试必须完全按顺序进行，且测试之间不得暂停。在所有测试期间，IUT的应答必须在由**Treply\_delay**指定的时间前返回，帧间间隔不得大于由**Tframe\_gap**定义的时间间隔，并在收到令牌(Token)或者主节点轮询帧(Poll For Master frame )后，IUT必须在由**Tusage\_delay**定义的时间间隔内开始传输下一帧。

1. 测试设置

若可能，应该将IUT设置为MAC地址2，并且将**Nmax\_master**设置为127，将**Nmax\_info\_frames**设置为2（若不能这样设置，则DONE\_WITH\_TOKEN:SendAnotherFrame的转换不可测试）。TD设置为MAC地址3。

TD必须知道被测设备的设备实例；并且必须知道该被测设备支持的所有的未确认服务。

1. 开始测试

这些测试验证IUT的基本操作。

1. SendFrame测试

目的：验证SendFrame的过程。此测试验证以下状态机的转换：

主节点：

INITIALIZING: DoneInitializing

IDLE: ReceivedPFM

接收帧：

IDLE: Preamble1

PREAMBLE: Preamble2

HEADER: FrameType, Destination, Source, Length1, Length2, HeaderCRC

HEADER CRC: Data (destination address = TS)

DATA: DataOctet, CRC1, CRC2

DATA CRC: GoodCRC

所需测试：无。

BACnet参考条款:9.3.2、9.5.5和9.5.6。

测试步骤：在本测试中，当IUT发出一个应答主节点轮询帧 (Reply To Poll For Master frame)时，被定义为完成初始化过程。

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

Poll For Master

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

Reply To Poll For Master

1. 服务请求转换

目的：验证IUT能够接收并正确理解已定义的帧，并且对应地创建正确格式的帧应答。此测试将验证以下的状态机的转换。

主节点：

INITIALIZING: DoneInitializing

IDLE: ReceivedDataNeedingReply

ANSWER DATA REQUEST: Reply

接收帧：

IDLE: Preamble1

PREAMBLE: Preamble2

HEADER: FrameType, Destination, Source, Length1, Length2, HeaderCRC

HEADER CRC: Data (destination address = TS)

DATA: DataOctet, CRC1, CRC2

DATA CRC: GoodCRC

BACnet参考条款：9.3.4、9.3.5、9.5.4、9.5.6。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request,

'Length' = (value from 1 to 50),

'Data' = ('Length' number of octets)

4. RECEIVE

Test\_Response

1. 状态机错误转换测试

本条款定义了在接收到确认和未确认的服务请求以及状态机遇到数据链路错误的情况下，证明接收帧状态机正确运行所需的测试用例。

所需测试：无。

1. 应答错误测试

本条款定义了数据链路出现错误导致帧被丢弃，且不发出应答的测试用例。

BACnet参考条款：9.5.4。

1. 错误数据CRC

目的：验证接收帧状态机是否检测并拒绝有错误数据CRC的帧。设X为IUT设备对象的实例号。该测试将验证以下接收帧状态机转换：

DATA CRC: BadCRC

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device,X),

'Property Identifier' = Object\_Identifier,

'Data CRC' = (any incorrect value)

6. BEFORE (**Acknowledgement Fail Time**) {

IF (ReadProperty-ACK received) THEN

ERROR "Incorrect MS/TP Frame Data CRC undetected."

}

7. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier,

'Service Request' = ReadProperty-Request

8. RECEIVE

ReadProperty-ACK,

'Frame Type' = BACnet Data Not Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier,

'Property Value' = (Device, X)

1. 数据超时

目的：验证接收帧状态机是否检测并拒绝数据字段传输期间超时的帧。该测试将验证以下接收帧状态机转换：

DATA: Timeout

测试步骤：在发送步骤5的过程中，应在发送CRC报头八比特字节组之后，在发送最终的数据CRC八比特字节组之前的时间段内，暂时停止传输一段时间（停止时间应大于**Tframe\_abort**）。

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

Test\_Request,

'Length' = (x where 1≤ x ≤50),

'Data' = (x number of octets)

6. BEFORE (**Treply\_delay**) {

IF (anything received) THEN

ERROR "MS/TP Framing error undetected."

}

7. TRANSMIT

Test\_Request,

'Length' = (x where 1≤ x ≤50),

'Data' = (x number of octets)

8. RECEIVE

Test\_Response,

'Data' = (the same data transmitted in step 7)

1. 数据成帧错误

目的：验证接收帧状态机能检测并拒绝在数据中有传输成帧错误的帧。该测试将验证以下接收帧状态机转换：

DATA: Error

BACnet参考条款：9.5.1.3。

测试步骤：在发送步骤5的过程中，应在数据位中的一个八位位组的停止位位置发送一个具有逻辑零的字节。

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

Test\_Request,

'Length' = (x where 1≤ x ≤50),

'Data' = (x number of octets)

6. BEFORE (**Treply\_delay**) {

IF (anything received) THEN

ERROR "MS/TP Framing error undetected."

}

7. TRANSMIT

Test\_Request,

'Length' = (x where 1≤ x ≤50),

'Data' = (x number of octets)

8. RECEIVE

Test\_Response

1. 帧头CRC错误

目的： 验证接收帧状态机是否检测并拒绝接收具有错误帧头CRC (Header CRC)的帧。该测试将验证以下接收帧状态机转换：

HEADER CRC: BadCRC

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

Test\_Request

'Header CRC' = (any incorrect value)

6. BEFORE (**Treply\_delay**) {

IF (anything received) THEN

ERROR "MS/TP Frame Header CRC error undetected."

}

7. TRANSMIT

Test\_Request

8. RECEIVE

Test\_Response

1. 非己方帧

目的：验证接收帧状态机检测并拒绝接收目标地址与IUT地址不同的帧。该测试将验证以下接收帧状态机转换：

HEADER CRC: NotForUs

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

DA = (MAC address not used by TD or IUT),

Test\_Request

6. BEFORE (**Treply\_delay**) {

IF (anything received) THEN

ERROR "Device responded to MAC address not its own."

}

7. TRANSMIT

Test\_Request

8. RECEIVE

Test\_Response

1. Header成帧错误

目的：验证接收帧状态机能检测并拒绝在Header中有传输成帧错误的帧。该测试将验证以下接收帧状态机转换：

HEADER: Error

BACnet参考条款：9.5.1.3。

测试步骤：在第5步传输中，帧头中停止比特位置的八位字节应被置为逻辑”0”再传输。

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

Test\_Request

6. BEFORE (**Treply\_delay**) {

IF (anything received) THEN

ERROR "MS/TP Frame Header framing error undetected."

}

7. TRANSMIT

Test\_Request

8. RECEIVE

Test\_Response

1. 帧头超时

目的：验证接收帧状态机是否检测并拒绝在帧头传输期间超时的帧。该测试将验证以下接收帧状态机转换：

HEADER: Timeout

测试步骤：在步骤5的发送期间，传输第二个前导码八位字节之后，在传输帧头CRC字节之前，停止发送。

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

Test\_Request

6. BEFORE (**Treply\_delay**) {

IF (anything received) THEN

ERROR "MS/TP Frame Header timeout undetected."

}

7. TRANSMIT

Test\_Request

8. RECEIVE

Test\_Response

1. 错误前导码

目的：验证接收帧状态机能准确拒绝接收格式错误的前导码。该测试将验证以下接收帧状态机转换：

HEADER: NotPreamble

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

X'55', (any value other than X'55' and X'FF')

6. TRANSMIT

Test\_Request

7. RECEIVE

Test\_Response

1. 出错

目的：验证接收帧状态机能正确地拒绝有接收错误发生的起始前导码字节。该测试将验证以下接收帧状态机转换：

IDLE EatAnError

测试步骤：在发送步骤5的过程中，前导码中的X'55'字节必须在停止位位置以逻辑零发送。

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

Test\_Request

6. BEFORE (**Treply\_delay**) {

IF (anything received) THEN

ERROR "MS/TP Preamble receive error undetected."

}

7. TRANSMIT

Test\_Request

8. RECEIVE

Test\_Response

1. 八位字节

目的：验证接收帧状态机能正确地拒绝没有值X'55'的初始前导码八位字节。该测试将验证以下接收帧状态机转换：

IDLE: EatAnOctet

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

Test Request,

'Preamble' = (any value other than X'55), X'FF'

6. BEFORE (**Treply\_delay**) {

IF (anything received) THEN

ERROR "MS/TP Frame incorrect Preamble undetected."

}

7. TRANSMIT

Test\_Request

8. RECEIVE

Test\_Response

1. 帧过长

目的：验证接收帧状态机是否对IUT帧过长的进行错误检查。该测试将验证以下接收帧状态机转换：

HEADER CRC: FrameTooLong

BACnet参考条款：9.5.4.4。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

Test\_Request,

'Length' = 502,

'Data' = (502 octets)

6. BEFORE (**Treply\_delay**) {

IF (anything received) THEN

ERROR "MS/TP Frame Header framing error undetected."

}

7. TRANSMIT

Test\_Request,

'Length' = 0

8. RECEIVE

Test\_Reply

1. 应答测试

本条款定义了纠正数据链路错误或因数据链路错误导致发出应答的测试用例。

1. 重复前导码1

目的：验证接收帧状态机是否会检测重复第一前导字节。该测试将验证以下接收帧状态机转换：

PREAMBLE: RepeatedPreamble1

BACnet参考条款：9.5.4.2。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

X'55'

6. TRANSMIT

Test\_Request

7. RECEIVE

Test\_Response

1. 空白帧测试请求

目的：验证接收帧状态机是否会接收空白测试请求帧,并通过Test\_Response应答。该测试将验证以下接收帧状态机转换：

HEADER CRC: NoData

BACnet参考条款：9.5.4.4。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request,

'Length' = 0

4. RECEIVE

Test\_Response,

'Length' = 0

1. 带数据的测试请求

目的：验证接收帧状态机是否会接收带有数据的测试请求帧，并通过Test\_Response应答。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request,

'Length' = (x where 1 ≤ x ≤ 50),

'Data' = (x number of octets)

4. RECEIVE

Test\_Response,

'Length' = (0 or value in step 3),

'Data' = (empty or octets transmitted in step 3)

1. 令牌操作的状态机转换测试

本条款定义了在接收、传输令牌以及前一个令牌丢失而生成一个新的令牌时，主节点有限状态机正确运行所需的测试用例。

1. 令牌传输至IUT

目的：验证IUT能正确接收并使用令牌，然后开始搜索下一个主节点。根据IUT的实现和设置，在已验证的主节点有限状态机转换序列中有三种可能：

IDLE: ReceivedToken (tested in all cases)

USE\_TOKEN: NothingToSend (tested if no data frames are sent)

USE\_TOKEN: SendNoWait (tested if a frame not expecting reply is sent)

DONE\_WITH\_TOKEN: SendAnotherFrame (tested if multiple frames are sent)

USE\_TOKEN: SendAndWait (tested if a frame expecting reply is sent)

WAIT\_FOR\_REPLY: ReplyTimeout (tested if there is no response to a frame expecting reply)

DONE\_WITH\_TOKEN: SendMaintenancePFM (tested in all cases)

BACnet 参考条款：9.3.1, 9.3.3, 9.5.6.5。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

Poll For Master

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

Reply To Poll For Master

4. TRANSMIT

Token

5. WHILE (received frame not a Poll For Master) DO {

}

6. RECEIVE

DA = IUT+1,

Poll For Master

1. 令牌通过IUT

目的：通过传递令牌来验证IUT是否正确响应了对主机轮询的回复。该测试将验证以下主节点有限状态机转换：

POLL\_FOR\_MASTER: ReceivedReplyToPFM

根据IUT的实现和设置，在已验证的主节点有限状态机转换序列中有三种可能：

在所有情形中：

PASS\_TOKEN: SawTokenUser

IDLE: ReceivedToken

情形一：无传输。

USE\_TOKEN: NothingToSend

情形二：传输一个不期望得到应答的帧

USE\_TOKEN: SendNoWait

这可以通过以下返回USE\_TOKEN状态的转换重复一次。

Done\_WITH\_TOKEN: SendAnotherFrame

情形三：传输一个期望得到应答的帧

USE\_TOKEN: SendAndWait

WAIT\_FOR\_REPLY: InvalidFrame

在所有情形中：

Done\_WITH\_TOKEN: SendToken (b)

BACnet参考条款：9.5.6.5，9.5.6.8。

所需测试：无。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

Poll For Master

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

Reply To Poll For Master

4. TRANSMIT

Token

5. WHILE (received other than Poll For Master frame) DO {

IF (frame is BACnet Data Expecting Reply) THEN

TRANSMIT

SA = (DA of received frame),

BACnet Data Not Expecting Reply,

'Header CRC' = (any incorrect value)

}

6. RECEIVE

DA = IUT+1,

Poll For Master

7. TRANSMIT

SA = IUT+1,

DA = IUT,

Reply To Poll For Master

8. RECEIVE

DA = IUT+1,

Token

9. TRANSMIT

DA = (MAC address other than DA and IUT),

Test\_Request

10. BEFORE (**Tno\_token** – 1 milliseconds) {

IF (anything received) THEN

ERROR "Passed token use undetected by IUT."

}

1. 传递后丢弃令牌

目的：当令牌在传递给另一个设备后被丢弃时，验证IUT的操作是否正确。该测试将验证如下状态转换。

PASS\_TOKEN: RetrySendToken

PASS\_TOKEN: FindNewSuccessor

POLL\_FOR\_MASTER: SendNextPFM (a)

BACnet参考条款：9.5.6.6。

所需测试：无。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

Poll For Master

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

Reply To Poll For Master

4. TRANSMIT

Token

5. WHILE (received frame other than Poll For Master) DO {

IF (frame is BACnet Data Expecting Reply) THEN

TRANSMIT

SA = (DA of received frame),

BACnet Data Not Expecting Reply,

'Header CRC' = (any incorrect value)

}

6. RECEIVE

DA = IUT+1,

Poll For Master

7. TRANSMIT

SA = IUT+1,

DA = IUT,

Reply To Poll For Master

8. RECEIVE

DA = IUT+1,

Token

9. WAIT **Tusage\_timeout**

10. RECEIVE

DA = IUT+1,

Token

11. WAIT **Tusage\_timeout**

12. RECEIVE

DA = IUT+2,

Poll For Master

13. WAIT **Tusage\_timeout**

14. RECEIVE

DA = IUT+3,

Poll For Master

1. 主节点轮询–无效帧

目的：当应答发送的主节点轮询帧，接收到无效帧时，验证IUT的操作是否正确。该测试将验证如下状态转换：

POLL\_FOR\_MASTER: SendNextPFM (b)

BACnet参考条款：9.5.6.8。

所需测试：无。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

Poll For Master

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

Reply To Poll For Master

4. TRANSMIT

Token

5. WHILE (received frame other than Poll For Master) DO {

IF (frame is BACnet Data Expecting Reply) THEN

TRANSMIT

SA = (DA of received frame),

BACnet Data Not Expecting Reply,

'Header CRC' = (any incorrect value)

}

6. RECEIVE

DA = IUT+1,

Poll For Master

7. TRANSMIT

SA = IUT+1,

DA = IUT,

Reply To Poll For Master

'Header CRC' = (any incorrect value)

8. RECEIVE

DA = IUT+2,

Poll For Master

1. 令牌接收与传递

目的：验证将令牌传递给一个主机MAC地址，而不是比IUT MAC地址高一个计数的地址。根据IUT的实现和设置，在已验证的主节点有限状态机转换序列中有三种可能：

在所有情形中：

PASS\_TOKEN: SawTokenUser

IDLE: ReceivedToken

情形一：无传输。

USE\_TOKEN: NothingToSend

情形二：传输一个不期望得到应答的帧

USE\_TOKEN: SendNoWait

这可以通过以下返回USE\_TOKEN状态的转换重复一次。

Done\_WITH\_TOKEN: SendAnotherFrame

情形三：传输一个期望得到应答的帧

USE\_TOKEN: SendAndWait

WAIT\_FOR\_REPLY: ReceivedPostpone

这可以通过以下返回USE\_TOKEN状态的转换重复一次。

Done\_WITH\_TOKEN: SendAnotherFrame

在所有情形中：

Done\_WITH\_TOKEN: SendToken (a)

BACnet参考条款：9.5.6.3、9.5.6.5。

所需测试：无。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

SA = 6,

Poll For Master

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

DA = 6,

Reply To Poll For Master

4. TRANSMIT

SA = 6,

Token

5. WHILE (received other than Token frame) DO {

IF (frame is BACnet Data Expecting Reply) THEN

TRANSMIT

SA = (DA of received frame),

BACnet Data Not Expecting Reply,

'Header CRC' = (any incorrect value)

}

6. RECEIVE

DA = 6,

Token

1. 完成轮询-未应答

目的：当最近一次主节点轮询帧没有应答时，在主节点轮询的周期结束时验证令牌的传递。根据IUT的实现和设置，在已验证的主节点有限状态机转换序列中有三种可能，在测试循环结束时还有另外两种选择：

在所有情形中：

PASS\_TOKEN: SawTokenUser

IDLE: ReceivedToken

情形一：无传输。

USE\_TOKEN: NothingToSend

情形二：传输一个不期望得到应答的帧

USE\_TOKEN: SendNoWait

这可以通过以下返回USE\_TOKEN状态的转换重复一次。

Done\_WITH\_TOKEN: SendAnotherFrame

情形三：传输一个期望得到应答的帧

USE\_TOKEN: SendAndWait

WAIT\_FOR\_REPLY: ReceivedPostpone

这可以通过以下返回USE\_TOKEN状态的转换重复一次。

Done\_WITH\_TOKEN: SendAnotherFrame

在所有情形中：

Done\_WITH\_TOKEN: SendToken (a)

或者

Done\_WITH\_TOKEN: SendMaintenancePFM

POLL\_FOR\_MASTER: DoneWithPFM (a)

BACnet参考条款：9.3.8、9.5.6.5。

所需测试：无。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

SA = 6,

Test Request

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

DA = 6,

Test Response

4. REPEAT I = (1 to 49) DO {

TRANSMIT

DA = 127,

SA = 6,

Poll For Master

WAIT **Tusage\_timeout**

TRANSMIT

SA = 6,

Token

WHILE (received other than Token frame) DO {

IF (frame is BACnet Data Expecting Reply) THEN

TRANSMIT

SA = (DA of received frame),

BACnet Data Not Expecting Reply,

'Header CRC' = (any incorrect value)

}

RECEIVE

DA = 6,

Token

}

5. TRANSMIT

DA = 127,

SA = 6,

Poll For Master

6. WAIT **Tusage\_timeout**

7. TRANSMIT

SA = 6,

Token

8. WHILE (received other than Token frame) DO {

IF (frame is BACnet Data Expecting Reply) THEN

TRANSMIT

SA = (DA of received frame),

BACnet Data Not Expecting Reply,

'Header CRC' = (any incorrect value)

}

9. RECEIVE

DA = 3,

Poll For Master

10. WAIT **Tusage\_timeout**

11. RECEIVE

DA = 6,

Token

1. 完成轮询-无效应答

目的：当最近一次主节点轮询帧是无效应答时，主节点轮询的周期结束时验证令牌的传递。已验证的主节点有限状态机转换序列除了最后转换不同外，其他的转换均与13.1.1.4.6相同:

POLL\_FOR\_MASTER: DoneWithPFM (b)

BACnet参考条款：9.5.6.8。

所需测试：无。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

SA = 6,

Test Request

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

DA = 6,

Test Response

4. REPEAT I = (1 to 49) DO {

TRANSMIT

DA = 127,

SA = 6,

Poll For Master

WAIT **Tusage\_timeout**

TRANSMIT

SA = 6,

Token

WHILE (received other than Token frame) DO {

IF (frame is BACnet Data Expecting Reply) THEN

TRANSMIT

SA = (DA of received frame),

BACnet Data Not Expecting Reply,

'Header CRC' = (any incorrect value)

}

RECEIVE

DA = 6,

Token

}

5. TRANSMIT

DA = 127,

SA = 6,

Poll For Master

6. WAIT **Tusage\_timeout**

7. TRANSMIT

SA = 6,

Token

8. WHILE (received other than Token frame) DO {

IF (frame is BACnet Data Expecting Reply) THEN

TRANSMIT

SA = (DA of received frame),

BACnet Data Not Expecting Reply,

'Header CRC' = (any incorrect value)

}

9. RECEIVE

DA = 3,

Poll For Master

10. WAIT **Tusage\_timeout**

11. RECEIVE

DA = 6,

Token

12. REPEAT I = (1 to 49) DO {

TRANSMIT

DA = 127,

SA = 6,

Poll For Master

WAIT **Tusage\_timeout**

TRANSMIT

SA = 6,

Token

WHILE (received other than Token frame) DO {

IF (frame is BACnet Data Expecting Reply) THEN

TRANSMIT

SA = (DA of received frame),

BACnet Data Not Expecting Reply,

'Header CRC' = (any incorrect value)

}

RECEIVE

DA = 6,

Token

}

13. TRANSMIT

DA = 127,

SA = 6,

Poll For Master

14. WAIT **Tusage\_timeout**

15. TRANSMIT

SA = 6,

Token

16. WHILE (received other than Token frame) DO {

IF (frame is BACnet Data Expecting Reply) THEN

TRANSMIT

SA = (DA of received frame),

BACnet Data Not Expecting Reply,

'Header CRC' = (any incorrect value)

}

17. RECEIVE

DA = 4,

Poll For Master

18. TRANSMIT

SA=4,

Reply To Poll For Master,

'Header CRC' = (any invalid value)

19. RECEIVE

DA = 6,

Token

1. 重置主节点轮询

目的：验证当轮询MAC地址小于下一个已知主机的MAC地址时，是否发生ResetMaintenancePFM转换。除了最后转换不同外，其他的转换均与12.1.1.4.7相同。

Done\_WITH\_TOKEN: ResetMaintenancePFM

BACnet参考条款：9.5.6.5。

所需测试：无。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

SA = 6,

Test Request

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

DA = 6,

Test Response

4. REPEAT M = (3 to 6) DO {

REPEAT I = (1 to 49) DO {

TRANSMIT

DA = 127,

SA = 6,

Poll For Master

WAIT **Tusage\_timeout**

TRANSMIT

SA = 6,

Token

WHILE (received other than Token frame) DO {

IF (frame is BACnet Data Expecting Reply) THEN

TRANSMIT

SA = (DA of received frame),

BACnet Data Not Expecting Reply,

'Header CRC' = (any incorrect value)

}

RECEIVE

DA = 6,

Token

}

TRANSMIT

DA = 127,

SA = 6,

Poll For Master

WAIT **Tusage\_timeout**

TRANSMIT

SA = 6,

Token

WHILE (received other than Token frame) DO {

IF (frame is BACnet Data Expecting Reply) THEN

TRANSMIT

SA = (DA of received frame),

BACnet Data Not Expecting Reply,

'Header CRC' = (any incorrect value)

}

IF ( M = 6) THEN

RECEIVE

DA = 3,

Poll For Master

ELSE

RECEIVE

DA = M,

Poll For Master

}

1. 下一个主机消失

目的：当最后一个已知的“下一个主设备” 没有收到或使用传递给它的令牌时，验证IUT在该“下一个主设备”MAC地址加1的MAC地址上能正确地恢复对主机的轮询。该测试验证如下转换：

PASS\_TOKEN: RetrySendToken

PASS\_TOKEN: FindNewSuccessor

BACnet参考条款: 9.5.6.6.

所需测试：无。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

SA = 6,

Poll For Master

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

DA = 6,

Reply To Poll For Master

4. TRANSMIT

SA = 6,

Token

5.WHILE (received other than Token frame) DO {

IF (frame is BACnet Data Expecting Reply) THEN

TRANSMIT

SA = (DA of received frame),

BACnet Data Not Expecting Reply,

'Header CRC' = (any incorrect value)

}

6. RECEIVE

DA = 6,

Token

7. WAIT **Tusage\_timeout**

8. RECEIVE

DA = 7,

Poll For Master

1. 主机轮询的应答帧-不正确的地址

目的:当IUT在主机轮询(Poll For Master)期间，收到具有错误目标地址的主机轮询应答帧时，验证IUT正确地转换到IDLE状态。该测试验证如下转换:

POLL\_FOR\_MASTER: ReceivedUnexpectedFrame (a)

IDLE: ReceivedPFM

BACnet参考条款: 9.5.6.8.

所需测试：无。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

SA = 6,

Poll For Master

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

DA = 6,

Reply To Poll For Master

4. TRANSMIT

SA = 6,

Token

5.WHILE (received other than Token frame) DO {

IF (frame is BACnet Data Expecting Reply) THEN

TRANSMIT

SA = (DA of received frame),

BACnet Data Not Expecting Reply,

'Header CRC' = (any incorrect value)

}

6. RECEIVE

DA = 6,

Token

7. WAIT **Tusage\_timeout**

8. RECEIVE

DA = 7,

Poll For Master

9. TRANSMIT

DA = (value less than 128 and other than IUT, 6 or 7),

Reply To Poll For Master

10. TRANSMIT

SA = 0,

Poll For Master

11. RECEIVE

DA = 0,

Reply To Poll For Master

1. 生成令牌

目的: 在丢失令牌时，验证IUT能生成令牌。该测试验证如下转换:

IDLE: LostToken

NO\_TOKEN: GenerateToken

BACnet参考条款: 9.5.6.7.

所需测试：无。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

SA = 6,

Poll For Master

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

DA = 6,

Reply To Poll For Master

4. RECEIVE

DA = IUT+1,

Poll For Master

1. 主机轮询 - 不正确的应答

目的:当在主机轮询期间收到非主机轮询应答帧时，验证IUT正确地转换到IDLE状态。该测试验证如下转换:

POLL\_FOR\_MASTER: ReceivedUnexpectedFrame (b)

IDLE: Received PFM

BACnet参考条款:9.5.6.7, 9.5.6.8.

所需测试：无。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

SA = 6,

Poll For Master

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

DA = 6,

Reply To Poll For Master

4. RECEIVE

Poll For Master

5. TRANSMIT

Test Request

6. TRANSMIT

SA = 0,

Poll For Master

7. RECEIVE

DA = 0,

Reply To Poll For Master

1. 可见帧

目的: 验证在丢弃令牌并由另一个设备创建新的令牌之后，IUT观察到此令牌的创建。

BACnet参考条款: 9.5.6.7.

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

SA = 6,

Poll For Master

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

DA = 6,

Reply To Poll For Master

4. WAIT (**Tno\_token** + **Tslot**)

5. TRANSMIT

DA = 127,

SA = 1,

Poll For Master

6. BEFORE (**Tno\_token** + **Tslot** ) {

IF (anything received) THEN

ERROR "Token incorrectly generated by IUT."

}

7. TRANSMIT

SA = 0,

Poll For Master

8. RECEIVE

DA = 0,

Reply To Poll For Master

1. 数据请求应答的验证测试

本条款描述了两个测试，用于验证与ANSWER\_DATA\_REQUEST状态关联的转换的正确操作。由于选择应答(Reply)还是延迟应答(Deferred Reply)是IUT内部的事情，因此可能无法确保完整的测试。如果选择是由外部因素可靠地确定的，例如读取属性的选择，则这些测试应执行两次，一次用于立即应答，一次用于延迟应答，以验证所有转换。设X为IUT设备对象的实例号。

下两个测试只需要通过其中一个即可。

1. 数据请求应答

目的: 验证针对BACnet Data Expecting Reply帧的正确应答。需验证的转换包括:

IDLE: ReceivedDataNeedingReply

ANSWER\_DATA\_REQUEST: Reply

BACnet参考条款:9.5.6.9.

所需测试：无。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

Test Request

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

Test Response

4. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'ObjectIdentifier' = (Device, X),

'PropertyIdentifier' = Object\_Identifier

5. RECEIVE

ReadProperty-ACK,

'Frame Type' = BACnet Data Not Expecting Reply,

'ObjectIdentifier' = (Device, X),

'PropertyIdentifier' = Object\_Identifier,

'Data' = (Device, X)

1. 延迟应答

只有在13.1.1.5.1中当IUT的应答为Reply Postponed帧时，才执行此测试。

目的:如果收到Reply Postponed帧，验证实际应答能正确。需测试的转换为:

IDLE: ReceivedDataNeedingReply

ANSWER\_DATA\_REQUEST: Deferred Reply

BACnet参考条款: 9.5.6.9.

所需测试: 数据请求应答, 13.1.1.5.1.

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

Test Request

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

Test Response

4. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'ObjectIdentifier' = (Device, X),

'PropertyIdentifier' = Object\_Identifier

5. RECEIVE

Reply Postponed

6.WHILE (received frame other than BACnet Data Not Expecting Reply) DO {

IF (frame type is Poll For Master) THEN

TRANSMIT

Reply To Poll For Master

ELSE

IF (frame type is Token) THEN

TRANSMIT

Token

}

7. RECEIVE

ReadProperty-ACK,

'Frame Type' = BACnet Data Not Expecting Reply,

'ObjectIdentifier' = (Device, X),

'PropertyIdentifier' = Object\_Identifier,

'Data' = (Device, X)

1. 其他的无应答测试

本条款描述的测试用于验证不会导致IUT应答的各种基本操作。通过测试验证主节点有限状态机仍处于IDLE状态。设X为IUT设备对象的实例号。

1. 收到数据未应答

目的：验证处于IDLE状态的主节点有限状态机能正确地处理不期望应答的BACnet数据(BACnet Data Not Expecting Reply)帧。需验证如下转换:

IDLE: ReceivedDataNoReply

IDLE: Received PFM

BACnet参考条款: 9.5.6.2.

所需测试：无。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

Test Request

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

Test Response

4. TRANSMIT

TimeSynchronization-Request,

'Frame Type' = BACnet Data Not Expecting Reply,

5.BEFORE (Treply\_delay) {

IF (anything received) THEN

ERROR "Response issued by IUT to BACnet Data Not Expecting Reply."

}

6. TRANSMIT

Poll For Master

7. RECEIVE

Reply To Poll For Master

1. 收到无效帧

目的:验证处于IDLE状态的主节点有限状态机能正确处理无效帧。需验证下列转换:

IDLE: ReceivedInvalidFrame

IDLE: Received PFM

BACnet参考条款: 9.5.6.2.

所需测试：无。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

Test Request

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

Test Response

4. TRANSMIT

Poll For Master,

'Header CRC'= (any incorrect value)

5. BEFORE (**Treply\_delay**) {

IF (frame received) THEN

ERROR "Invalid Frame accepted by IUT."

}

6. TRANSMIT

Poll For Master

7. RECEIVE

Reply To Poll For Master

1. 无用帧的测试

本条款描述的测试用于验证处于IDLE状态的IUT拒绝未寻址到它的帧或非法广播的帧。第13.1.1.6.3.1条款验证的转换不会发生，因为接收帧状态机在其NotForUs转换中会拒绝那些发给其他设备的消息。所验证的转换如下：

IDLE: ReceivedUnwantedFrame

IDLE: Received PFM

BACnet参考条款: 9.5.6.2.

所需测试：无。

1. 非己方地址

目的:验证处于IDLE状态的IUT能正确地处理指向另一个设备的帧。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

Test Request

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

Test Response

4. TRANSMIT

DA = (value less than 128 and other than IUT or TD),

Poll For Master

5.BEFORE (**Treply\_delay**) {

IF (frame received) THEN

ERROR "IUT accepted frame addressed to other device."

}

6. TRANSMIT

Poll For Master

7. RECEIVE

Reply To Poll For Master

1. 令牌帧广播

目的:验证处于IDLE状态的IUT能正确地处理令牌帧广播。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

Test Request

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

Test Response

4. TRANSMIT

DA = LOCAL BROADCAST,

Token

5.BEFORE (**Treply\_delay**) {

IF (frame received) THEN

ERROR "Broadcast Token frame accepted by IUT."

}

6. TRANSMIT

Poll For Master

7. RECEIVE

Reply To Poll For Master

1. BACnet数据期望回复帧广播

目的:验证处于IDLE状态的IUT能正确地处理不期望应答的BACnet数据帧的广播。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

Test Request

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

Test Response

4. TRANSMIT

DA = LOCAL BROADCAST,

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'ObjectIdentifier' = (Device, X),

'PropertyIdentifier' = Object\_Identifier

5. BEFORE (**Treply\_delay**) {

IF (frame received) THEN

ERROR "Broadcast BACnet Data Expecting Reply frame accepted by IUT."

}

6. TRANSMIT

Poll For Master

7. RECEIVE

Reply To Poll For Master

1. 测试请求帧广播

目的:验证处于IDLE状态的IUT能正确地处理测试请求帧广播。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

Test Request

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

Test Response

4. TRANSMIT

DA = LOCAL BROADCAST,

Test Request

5. BEFORE (**Treply\_delay**) {

IF (frame received) THEN

ERROR "Broadcast Test Request frame accepted by IUT."

}

6. TRANSMIT

Poll For Master

7. RECEIVE

Reply To Poll For Master

1. 唯一主机的测试

测试验证IUT能够正确地识别自己是MS/TP LAN上唯一的主机，并能识别另一个主机的加入。

1. 丢失令牌

目的:验证IUT能识别出令牌丢失并生成另一个令牌。该测试验证如下转换:

IDLE: LostToken

NO\_TOKEN: GenerateToken

BACnet参考条款: 9.5.6.7.

所需测试：无。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

Poll For Master

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

Reply To Poll For Master

4.BEFORE (**Tno\_token** + (2 \* **Tslot**)) {

IF (frame received) THEN

ERROR "Lost Token detected too early by IUT."

}

5. BEFORE (**Tslot**) RECEIVE

DA = IUT+1,

Poll For Master

1. 下一个主机轮询

目的: 验证IUT拥有令牌，并正在进行轮询以寻找另一个主机。这测试验证如下转换:

POLL\_FOR\_MASTER: SendNextPFM

BACnet参考条款: 9.5.6.8.

所需测试：无。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

Poll For Master

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

Reply To Poll For Master

4. RECEIVE

DA = IUT+1,

Poll For Master

5.BEFORE (**Tusage\_timeout**) {

IF (frame received) THEN

ERROR "IUT didn”t wait long enough for Reply To Poll For Master."

}

6. RECEIVE

DA = IUT+2,

Poll For Master

1. 更多轮询

目的: 验证IUT能检查其轮询中所有剩余的MAC地址以找到另一个主机。在最后一步，本测试转换:

POLL\_FOR\_MASTER: DeclareSoleMaster (a)

BACnet参考条款: 9.5.6.8.

所需测试：无。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

Poll For Master

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

Reply To Poll For Master

4. RECEIVE

DA = IUT+1,

Poll For Master

5.REPEAT X=(IUT+2 to 127, 0 to IUT-1) {

BEFORE (**Tusage\_timeout**) {

IF (frame received) THEN

ERROR "IUT didn’t wait long enough for Reply To Poll For Master."

}

RECEIVE

DA = X,

Poll For Master

}

1. 宣称唯一主机(a)

目的:验证IUT已宣布自己是唯一的主机，但仍在进行另一个主机的轮询。该测试验证下列转换:

DONE\_WITH\_TOKEN: Solemaster (a)

DONE\_WITH\_TOKEN: SendMaintenancePFM

BACnet参考条款: 9.5.6.5.

所需测试：无。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

Poll For Master

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

Reply To Poll For Master

4. RECEIVE

DA = IUT+1,

Poll For Master

5.REPEAT X=(IUT+2 to 127, 0 to IUT-1) {

BEFORE (Tusage\_timeout) {

IF (frame received) THEN

ERROR "IUT didn’t wait long enough for Reply To Poll For Master."

}

RECEIVE

DA = X,

Poll For Master

}

6. BEFORE (**Tusage\_timeout**) {

IF (frame received) THEN

ERROR "IUT didn’t wait long enough for Reply To Poll For Master."

}

7. RECEIVE

DA = IUT+1,

Poll For Master

1. 新主机加入

目的: 验证IUT能识别出另一个主机的存在。

BACnet参考条款: 9.5.6.8.

所需测试：无。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

Poll For Master

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

Reply To Poll For Master

4. RECEIVE

DA = IUT+1,

Poll For Master

5.REPEAT X=(IUT+2 to 127, 0 to IUT-1) {

BEFORE (**Tusage\_timeout**) {

IF (frame received) THEN

ERROR "IUT didn’t wait long enough for Reply To Poll For Master."

}

RECEIVE

DA = X,

Poll For Master

}

6. BEFORE (**Tusage\_timeout**) {

IF (frame received) THEN

ERROR "IUT didn’t wait long enough for Reply To Poll For Master."

}

7. RECEIVE

DA = IUT+1,

Poll For Master

8. TRANSMIT

SA = IUT+1,

Reply To Poll For Master

9. RECEIVE

DA = IUT+1,

Token

1. 下一个主机轮询

目的: 验证IUT拥有令牌，并正在进行轮询以寻找另一个主机。该测试验证下列转换:

POLL\_FOR\_MASTER: SendNextPFM

BACnet参考条款: 9.5.6.8.

所需测试：无。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

Poll For Master

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

Reply To Poll For Master

4. RECEIVE

DA = IUT+1,

Poll For Master

5.REPEAT X=(IUT+2 to 127, 0 to IUT-1) {

BEFORE (**Tusage\_timeout**) {

IF (frame received) THEN

ERROR "IUT didn’t wait long enough for Reply To Poll For Master."

}

RECEIVE

DA = X,

Poll For Master

}

6.BEFORE (**Tusage\_timeout**) {

IF (frame received) THEN

ERROR "IUT didn’t wait long enough for Reply To Poll For Master."

}

7. RECEIVE

DA = IUT+1,

Poll For Master

8. TRANSMIT

SA = IUT+1,

Reply To Poll For Master

9. RECEIVE

DA = IUT+1

Token

10. BEFORE (**Tusage\_timeout**) {

IF (frame received) THEN

ERROR "IUT didn’t wait long enough for Reply To Poll For Master."

}

11. RECEIVE

DA = IUT+2,

Poll For Master

1. 宣称唯一主机(b)

目的: 对所发送的主机轮询帧，IUT收到的应答是无效帧时，验证IUT可进行必要的转换，从而声明自己是唯一的主机。测试的转换包括:

POLL\_FOR\_MASTER: SendNextPFM

POLL\_FOR\_MASTER: DeclareSoleMaster (b)

BACnet参考条款: 9.5.6.8.

所需测试：无。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

Poll For Master

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

Reply To Poll For Master

4. RECEIVE

DA = IUT+1,

Poll For Master

5. TRANSMIT

Reply To Poll For Master

SA = IUT+1,

'Header CRC' = (any incorrect value)

6. REPEAT X=(IUT+2 to 127, 0 to IUT-1) {

RECEIVE

DA = X,

Poll For Master

TRANSMIT

SA = X,

Reply To Poll For Master,

'Header CRC' = (any incorrect value)

}

1. 唯一主机(b)

目的: 验证IUT已宣称自己是唯一的主机，并继续轮询其他主机。根据IUT的实现和设置，需验证的主节点有限状态机转换序列有三种可能，在测试循环的最后还有两种选择:

1. 什么也没有发送.

USE\_TOKEN: NothingToSend

2. 发送一个无需应答的帧:

USE\_TOKEN: SendNoWait

通过以下转换返回到这一步开头，此测试可以重复一次:

DONE\_WITH\_TOKEN: SendAnotherFrame

3. 发送一个需应答的帧:

USE\_TOKEN: SendAndWait

在所有情况下：

WAIT\_FOR\_REPLY: ReceivedReply

前49次:

DONE\_WITH\_TOKEN: SoleMaster

第50次：

DONE\_WITH\_TOKEN: SendMaintenancePFM

BACnet参考条款: 9.5.6.8.

所需测试：无。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

Poll For Master

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

Reply To Poll For Master

4. RECEIVE

DA = IUT+1,

Poll For Master

5. TRANSMIT

Reply To Poll For Master,

SA = IUT+1,

'Header CRC' = (any incorrect value)

6.REPEAT X=(IUT+2 to 127, 0 to IUT-1) {

RECEIVE

DA = X,

Poll For Master

TRANSMIT

SA = X,

Reply To Poll For Master,

'Header CRC' = (any incorrect value)

}

7. RECEIVE

DA = IUT+1,

Poll For Master

1. 获得令牌

目的: 验证IUT能正确地应答另一个主机的加入。该测试要验证下列转换：

POLL\_FOR\_MASTER: ReceivedReplyToPFM

BACnet参考条款: 9.5.6.8.

所需测试：无。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. WHILE (IUT not initialized) DO {

TRANSMIT

Poll For Master

IF (IUT is turned off) THEN

MAKE (IUT turned on or otherwise started)

}

3. RECEIVE

Reply To Poll For Master

4. RECEIVE

DA = IUT+1,

Poll For Master

5. TRANSMIT

Reply To Poll For Master,

SA = IUT+1,

'Header CRC'= (any incorrect value)

6. REPEAT X=(IUT+2 to 127, 0 to IUT-1) {

RECEIVE

DA = X,

Poll For Master

TRANSMIT

SA = X,

Reply To Poll For Master,

'Header CRC'= (any incorrect value)

}

7. RECEIVE

DA = IUT+1,

Poll For Master

8. TRANSMIT

SA = IUT+1,

Reply To Poll For Master

9. RECEIVE

DA = IUT+1,

Token

1. 在确认服务请求期间检测到多个令牌

本条款定义了仅当IUT能够周期性地发起确认服务请求(如ReadProperty请求)时才执行的测试。每个测试都等待至IUT生成一个令牌并使用它，然后TD传输一个无效的帧类型(指示另一个令牌的存在)，导致IUT重新进入IDLE状态，然后进行测试。这些测试验证的转换条件如下:

WAIT\_FOR\_REPLY: ReceivedUnexpectedFrame (a,b)

IUT应设置为向目的地MAC地址为3或更高的地址，发送重复确认的服务请求(即ReadProperty请求)。

设X为IUT设备对象的实例号。

BACnet参考条款: 9.5.6.4.

所需测试：无。

1. 不同的目的地

目的:当一个不同MAC地址的消息出现错误时，验证能检测到第二个令牌。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. MAKE (IUT to generate a confirmed service request)

6. WHILE (BACnet Data Expecting Reply frame not received) DO {

}

7. TRANSMIT

DA = (value less than 128 and other than IUT or TD),

BACnet Data Not Expecting Reply

8. WAIT **Treply\_timeout**

9. TRANSMIT

SA = 0,

DA = IUT,

Poll For Master

10. RECEIVE

DA = 0,

SA = IUT,

Reply To Poll For Master

1. 广播

目的:在广播消息出错时，验证能检测第二个令牌。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. MAKE (IUT to generate a confirmed service request)

6. WHILE (BACnet Data Expecting Reply frame not received) DO {

}

7. TRANSMIT

DA = LOCAL BROADCAST,

BACnet Data Not Expecting Reply

8. WAIT **Treply\_timeout**

9. TRANSMIT

SA = 0,

DA = IUT,

Poll For Master

10. RECEIVE

DA = 0,

SA = IUT,

Reply To Poll For Master

1. 令牌

目的:验证在第二个令牌传递给已经拥有令牌的IUT时能检测它。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. MAKE (IUT to generate a confirmed service request)

6. WHILE (BACnet Data Expecting Reply frame not received) DO {

}

7. TRANSMIT

Token

8. WAIT **Treply\_timeout**

9. TRANSMIT

SA = 0,

DA = IUT,

Reply To Poll For Master

10. RECEIVE

DA = 0,

SA = IUT,

Reply To Poll For Master

1. 主机轮询

目的:在轮询带有令牌的IUT时，检测第二个令牌。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. MAKE (IUT to generate a confirmed service request)

6. WHILE (BACnet Data Expecting Reply frame not received) DO {

}

7. TRANSMIT

SA = 0,

DA = IUT,

Poll For Master

8. WAIT **Treply\_timeout**

9. TRANSMIT

SA = 0,

DA = IUT,

Poll For Master

10. RECEIVE

DA = 0,

SA = IUT,

Reply To Poll For Master

1. 主机轮询应答

目的:在返回不正确的应答时检测协议问题。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. MAKE (IUT to generate a confirmed service request)

6. WHILE (BACnet Data Expecting Reply frame not received) DO {

}

7. TRANSMIT

SA = 0,

DA = IUT,

Reply To Poll For Master

8. WAIT **Treply\_timeout**

9. TRANSMIT

SA = 0,

DA = IUT,

Poll For Master

10. RECEIVE

DA = 0,

SA = IUT,

Reply To Poll For Master

1. 测试请求

目的:当带有令牌的IUT接收到Test\_Request帧时，检测第二个令牌。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. MAKE (IUT to generate a confirmed service request)

6. WHILE (BACnet Data Expecting Reply frame not received) DO {

}

7. TRANSMIT

SA = 0,

DA = IUT,

Test Request

8. WAIT **Treply\_timeout**

9. TRANSMIT

SA = 0,

DA = IUT,

Poll For Master

10. RECEIVE

DA = 0,

SA = IUT,

Reply To Poll For Master

1. 期待应答的BACnet数据

目的:在轮询带有令牌的IUT时，检测第二个令牌。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

1. RECEIVE

Test\_Response

5. MAKE (IUT to generate a confirmed service request)

6. WHILE (BACnet Data Expecting Reply frame not received) DO {}

7. TRANSMIT

SA = 0,

DA = IUT,

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'ObjectIdentifier' = (Device, X),

'PropertyIdentifier' = Object\_Identifier

8. WAIT **Treply\_timeout**

9. TRANSMIT

SA = 0,

DA = IUT,

Poll For Master

10. RECEIVE

DA = 0,

SA = IUT,

Reply To Poll For Master

1. 令牌使用测试

本条款定义了仅在IUT能够周期性地发起一个确认服务请求（如ReadProperty请求），或者能够周期性地应答另一个未确认服务请求（如I-Am请求）时，才执行的测试。

如果可以，应设置IUT以发送重复确认服务请求(即ReadProperty请求)，目的地MAC地址为3或更高。还应该将IUT设置为可发送未确认服务请求。

如果IUT的设备对象的Max\_Info\_Frames属性是可变的，那么它最初应该设置为1, Number\_of\_APDU\_Retries的值应该是2或者更大。

BACnet参考条款: 9.5.6.

所需测试：无。

1. 未确认请求

只有当IUT支持未确认服务Who-Is或Who-Has，并相应地对I-Am和I-Have服务应答时，才执行此测试。

目的: 验证主节点有限状态机对未确认的服务请求能正确执行。该测试验证下列转换:

IDLE LostToken

NO\_TOKEN GenerateToken

POLL\_FOR\_MASTER SendNextPFM

IDLE ReceivedDataNoReply

USE\_TOKEN SendNoWait

所需测试：无。

测试步骤:在这些步骤中，如果不支持Who-Is服务，则替换为Who-Has和I-Have服务。

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. WHILE (no Token frame received) DO {

IF (Poll For Master frame received with DA set to TD)THEN

TRANSMIT

Reply To Poll For Master

}

4. TRANSMIT

DA = LOCAL BROADCAST,

Who-Is-Request

5. TRANSMIT

Token

6. RECEIVE

I-Am-Request

1. 带应答的确认请求

只有当IUT能够执行确认服务请求时，才进行此测试。

目的: 验证主节点有限状态机正确执行确认服务请求。该测试验证下列转换:

USE\_TOKEN: SendAndWait

WAIT\_FOR\_REPLY: ReceivedReply

所需测试：无。

测试步骤:步骤6的ReadProperty-ACK是对步骤5请求的适当和正确的应答。

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. MAKE (IUT initiate ReadProperty-Request to TD)

4. WHILE (ReadProperty-Request with DA set to TD not received) DO {

IF (Poll For Master frame received with DA set to TD) THEN

TRANSMIT

Reply To Poll For Master

IF (Token frame received with DA set to TD not received) THEN

TRANSMIT

Token

}

5. RECEIVE

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

6. TRANSMIT

ReadProperty-ACK

7. BEFORE (APDU\_Timeout) {

IF (Poll For Master frame received with DA set to TD) THEN

TRANSMIT

Reply To Poll For Master

IF (Token frame received with DA set to TD not received) THEN

TRANSMIT

Token

IF (ReadProperty-Request identical to step 5 received) THEN

ERROR "Confirmed Service ACK not understood by IUT."

}

1. 确认请求-无应答

只有在IUT能够发起确认服务请求时，才进行此测试。

目的：在未收到应答时，验证主节点有限状态机能正确终止确认服务请求。该测试要验证以下转换：

USE\_TOKEN: SendAndWait

WAIT\_FOR\_REPLY: ReplyTimeout

所需测试：无。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned off or otherwise started)

3. MAKE (IUT initiate ReadProperty-Request to TD)

4. REPEAT X = (1 to IUT APDU\_Retry\_Count) DO {

BEFORE (APDU\_Timeout) {

IF (ReadProperty-Request with DA set to TD received) THEN

ERROR "Retry too soon."

IF (Poll For Master frame received with DA set to TD) THEN

TRANSMIT

Reply To Poll For Master

IF (Token frame received with DA set to TD not received) THEN

TRANSMIT

Token

}

RECEIVE

ReadProperty-Request

}

5. BEFORE (APDU\_Timeout) {

IF (ReadProperty-Request with DA set to TD received) THEN

ERROR "Incorrectly terminated service request."

IF (Poll For Master frame received with DA set to TD) THEN

TRANSMIT

Reply To Poll For Master

IF (Token frame received with DA set to TD not received) THEN

TRANSMIT

Token

}

1. 确认请求–无效应答

只有在IUT能够发起确认服务请求时，才进行此测试。

目的：验证主节点有限状态机能正确处理无效应答帧。该测试要验证以下转换：

USE\_TOKEN: SendAndWait

WAIT\_FOR\_REPLY: InvalidFrame

所需测试：无。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. MAKE (IUT initiate ReadProperty-Request to TD)

4. REPEAT X = (1 to IUT APDU\_Retries) DO {

BEFORE (APDU\_Timeout) {

IF (ReadProperty-Request with DA set to TD received) THEN

ERROR "Retry too soon."

IF (Poll For Master frame received with DA set to TD) THEN

TRANSMIT

Reply To Poll For Master

IF (Token frame received with DA set to TD not received) THEN

TRANSMIT

Token

}

RECEIVE

ReadProperty-Request

TRANSMIT

ReadProperty-ACK,

'Frame Type' = BACnet Data Expecting Reply,

'Data CRC' = (any incorrect value)

}

5. BEFORE (APDU\_Timeout) {

IF (ReadProperty-Request with DA set to TD received) THEN

ERROR "Incorrectly terminated service request."

IF (Poll For Master frame received with DA set to TD) THEN

TRANSMIT

Reply To Poll For Master

IF (Token frame received with DA set to TD not received) THEN

TRANSMIT

Token

}

1. 确认请求-延迟应答

只有在IUT能够发起确认服务请求时，才进行此测试。

目的：验证主节点有限状态机能正确终止延迟应答。该测试要验证以下转换：

USE\_TOKEN: SendAndWait

WAIT\_FOR\_REPLY: ReceivedPostponed

所需测试：无。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. MAKE (IUT initiate ReadProperty-Request to TD)

4. REPEAT X = (1 to IUT APDU\_Retries) DO {

BEFORE (APDU\_Timeout) {

IF (ReadProperty-Request with DA set to TD received) THE

ERROR "Retry too soon."

IF (Poll For Master frame received with DA set to TD) THEN

TRANSMIT

Reply To Poll For Master

IF (Token frame received with DA set to TD not received) THEN

TRANSMIT

Token

}

RECEIVE

ReadProperty-Request

TRANSMIT

Reply Postponed

}

5. BEFORE (APDU\_Timeout) {

IF (ReadProperty-Request with DA set to TD received) THEN

ERROR "Incorrectly terminated service request."

IF (Poll For Master frame received with DA set to TD) THEN

TRANSMIT

Reply To Poll For Master

IF (Token frame received with DA set to TD not received) THEN

TRANSMIT

Token

}

1. 最大信息帧检查

所需测试：无。

BACnet参考条款:9.5.3和9.5.6.5

目的：该检查验证在IUT收到令牌到传递此令牌或发起主机轮询的时间之间，MS/TP 主节点状态机发出的信息帧数不超过Nmax\_info\_frames。与测试不同，检查不是由测试步骤构成的，而是一些在完整的测试过程必须满足的条件。因此，检查在测试执行期间或执行之后要周期性地验证。

配置建议：如果设备对象的Max\_Info\_Frames属性是可配置的，那么建议将此属性设置为其最小设置，以便至少涉及到正在测试的MS/TP端口的某些测试性能。

检查条件：在执行期间监视MS/TP LAN，其中IUT可能会发出一些信息帧；如果IUT发出的信息帧数多于：

a) 在接收和传递令牌之间的时间间隔内的Max\_Info\_Frames配置值（局域网上有多个主机），或

b) 在接收令牌和发出PFM之间的时间间隔内的Max\_Info\_Frames配置值（局域网上有多个主机），或

c) 在任意两个连续主机轮询帧之间的时间间隔内的Max\_Info\_Frames配置值，但发送主机轮询到(TS-1) modulo Max\_Master和发送主机轮询到 (TS+1) modulo Max\_Master之间的时间间隔除外（IUT是局域网中唯一的主机），或

d)在发送主机轮询帧到(TS-1) modulo Max\_Master，和接下来的发送主机轮询帧到(TS+1) modulo Max\_Master 的时间间隔内，是Max\_Info\_Frames的配置值的52倍，（IUT是局域网中唯一的主机），

那么，该IUT将无法通过检查。

测试人员注意事项：在d)中使用的值为52，因为在标准135-1995中最初定义的 MS/TP状态机中有一个错误导致在主机轮询周期中令牌传递了52次，而不是50次。

1. +MS/TP从设备测试

该条款中定义的测试用于验证BACnet MS/TP从设备能正确实现Receive Frame有限状态机，SendFrame过程，和Slave Node有限状态机。而ANSWER DATA REQUEST：CannotReply，状态转换不可检测。

Slave Node有限状态机的部分转换不能验证，这是由于并行转换的存在和TD不能唤起特定的转换。如果在测试中观察到任意一个并行转换，将认为IUT与BACnet一致。

这些测试应在IUT支持的所有波特率下进行，且在MS/TP LAN上只有IUT和TD存在。

1. 对正常确认和未确认服务请求的状态机转换测试

该条款定义了在正确收到并处理了确认和未确认服务请求的环境中，验证Receive Frame状态机和Slave Node状态机能正确操作所必要的测试用例。设X为IUT设备对象的实例号数。

1. 确认服务请求转换

目的：验证IUT可以正确接收和理解形成的帧，并且在应答中能生成正确的帧。该测试验证以下的状态机转换。

Slave Node:

INITIALIZING: DoneInitializing

IDLE: ReceivedDataNeedingReply

ANSWER DATA REQUEST: Reply

Receive Frame:

IDLE: Preamble1

PREAMBLE: Preamble2

HEADER: FrameType,Destination,Source,Length1,Length2, HeaderCRC

HEADER CRC: Data (destination address = TS)

DATA: DataOctet, CRC1, CRC2

DATA CRC: GoodCRC

该测试也可以检测MS/TP中的05类型帧，BACnet Data Expecting Reply

所需测试：无。

BACnet参考条款：9.3.6, 9.5.5, 9.5.7 和9.5.4.

测试步骤：在该测试中返回的应答，帧间时隙应不大于**Tframe\_gap。**

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier,

6. BEFORE (**Treply\_delay**) RECEIVE

ReadProperty-ACK,

'Frame Type' = BACnet Data Not Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier,

'Property Value' = (Device, X)

1. 定向BACnet数据不需要应答

目的：验证对直接寻址的BACnet数据类型的帧没有应答，且Slave Node有限状态机保持在IDLE状态。该测试要验证以下Slave Node有限状态机转换：

IDLE: ReceivedDataNoReply

所需测试：无。

BACnet参考条款:9.5.7.2.

测试步骤：通过该检测如果IUT不支持任何未确认服务，该测试将不会被执行，并且IUT将被认为与BACnet的本条款内容具有一致性。

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned off or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

(Unconfirmed service supported by IUT),

'Frame Type' = BACnet Data Expecting Reply,

6. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier

7. BEFORE (**Treply\_delay**) RECEIVE

ReadProperty-ACK,

'Frame Type' = BACnet Data Not Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier,

'Property Value' = (Device, X)

1. 广播BACnet数据不需要应答

目的：验证没有对广播帧类型BACnet数据不期望应答的应答，且Slave Node有限状态机保持在IDLE状态。该测试要验证以下Slave Node有限状态机转换：

HEADER\_CRC: Data (destination = 255)

所需测试：确认服务请求转换，12.1.2.1.1.

BACnet参考条款：9.5.4.4.

测试步骤：如果IUT不支持任何广播未确认服务，该测试将不会被执行，并且IUT将被认为与BACnet本条款的内容具有一致性。

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

DA = LOCAL BROADCAST,

(Unconfirmed service supported by IUT),

'Frame Type' = BACnet Data Not Expecting Reply,

6. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier,

7. BEFORE (**Treply\_delay**) RECEIVE

ReadProperty-ACK,

'Frame Type' = BACnet Data Not Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier,

'Property Value' = (Device, X)

1. 错误转换下状态机转换测试

该条款定义了在Receive Frame状态机和Slave Node状态机收到确认和未确认服务请求且状态机数据链接错误时，验证状态机正确操作的必要测试。

所需测试：无。

1. 无应答错误测试

该条款定义了数据链路错误引起帧被丢弃和没有发出应答的测试用例。这些测试用例检测以下Slave Node有限状态机转换：

IDLE: ReceivedInvalidFrame

BACnet参考条款：9.5.4 and 9.5.7.2.

1. 数据CRC错误

目的：验证接收帧状态机能检测并拒绝具有错误数据CRC的帧。该测试验证如下的接收帧状态机转换：

DATA\_CRC: BadCRC

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier

'Data CRC' = (any incorrect value),

6. BEFORE (**Acknowledgement Fail Time**) {

IF (ReadProperty-ACK received) THEN

ERROR "Incorrect MS/TP Frame Data CRC undetected."

}

7. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier,

8. RECEIVE

ReadProperty-ACK,

'Frame Type' = BACnet Data Not Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier,

'Property Value' = (Device, X)

1. 数据超时

目的：验证接收帧状态机能检测并拒绝在数据域传输过程中超时的帧。该测试验证以下的接收帧状态机转换：

DATA: Timeout

检测步骤：在检测过程中的第5步，在Header CRC八位字节被传输后，在最后Data CRC八位字节被传输之前，停止传输。

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier

6. WAIT **Tframe\_abort**

7. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Type,

8. RECEIVE

ReadProperty-ACK,

'Frame Type' = BACnet Data Not Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

1. 数据成帧错误

目的：验证接收帧状态机能检测并拒绝数据中有传输成帧错误的帧。该测试要验证如下的接收帧状态机的转换：

DATA: Error

BACnet参考条款:9.5.1.3.

测试步骤：在该测试的第5步，数据域中停止比特位置的八位字节要填充逻辑零传输。

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier

6. WAIT **Treply\_delay**

7. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Type,

8. RECEIVE

ReadProperty-ACK,

'Frame Type' = BACnet Data Not Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

1. Header CRC 错误

目的：验证接收帧状态机能检测并拒绝带有错误Header CRC的帧。该测试要验证如下的接收帧状态机的转换：

HEADER\_CRC: BadCRC

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier

'Header CRC' = (any incorrect value),

6. WAIT **Treply\_delay**

7. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Type,

8. RECEIVE

ReadProperty-ACK,

'Frame Type' = BACnet Data Not Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

1. 非己方帧

目的：验证接收帧状态机能检测并拒绝目标地址异于IUT地址的帧，该测试要验证如下的接收帧状态机的转换：

HEADER\_CRC: NotForUs

因为接收帧状态机首先检查目标地址，以下Slave Node状态机转换不会出现，但该转换或可在其他地方检测。

IDLE ReceivedUnwantedFrame (a)

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

ReadProperty-Request,

DA = (value less than 128 and other than IUT and TD),

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier

6. WAIT **Treply\_delay**

7. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Type,

'Service Request' = ReadProperty-Request

8. RECEIVE

ReadProperty-ACK,

'Frame Type' = BACnet Data Not Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

1. Header 成帧错误

目的：验证接收帧状态机能检测并拒绝传输Header有成帧错误的帧。该测试要验证如下的接收帧状态机的转换：

HEADER: Error

BACnet参考条款:9.5.1.3.

测试步骤：第5步的传输中，Header中停止比特位置的八位字节要填充逻辑零后再传输。

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier

6. WAIT **Treply\_delay**

7. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Type,

8. RECEIVE

ReadProperty-ACK,

'Frame Type' = BACnet Data Not Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

1. Header 超时

目的：验证接收帧状态机能检测并拒绝在头部(header field)传输中超时的帧。该测试要验证如下的接收帧状态机的转换：

HEADER: Timeout

测试步骤：第5步的传输中，前导码的第二个八位字节传输后，在Header CRC八位字节传输前，停止传输。

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier

6. WAIT **Treply\_delay**

7. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Type,

8. RECEIVE

ReadProperty-ACK,

'Frame Type' = BACnet Data Not Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

1. 错误前导码

目的：验证接收帧状态机能正确拒绝错误的前导码。该测试要验证如下的接受帧状态机的转换：

HEADER: NotPreamble

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

ReadProperty-Request,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Type

4. RECEIVE

ReadProperty-ACK,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

5. TRANSMIT

X'55'

6. TRANSMIT

(octet other than X'55' and X'FF')

7. TRANSMIT

ReadProperty-Request,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Type

8. RECEIVE

ReadProperty-ACK,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

1. 出错

目的：验证接收帧状态机能正确地拒绝有接收错误的前导码起始字节。该测试要验证如下的接收帧状态机的的转换：

IDLE: EatAnError

测试步骤：在第5步的传输中，前导码在停止比特位置的X'55'字节要用逻辑零填充后再传输。

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier

6. WAIT **Treply\_delay**

7. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Type,

8. RECEIVE

ReadProperty-ACK,

'Frame Type' = BACnet Data Not Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

1. 八位字节

目的：验证接收帧状态机能正确地拒绝没有值X'55'的初始前导码的八位字节，该测试要验证如下的接收帧状态机的的转换：

IDLE: EatAnOctet

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

ReadProperty-Request,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Type

6. RECEIVE

ReadProperty-ACK,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

7. TRANSMIT

(octet other than X'55')

8. TRANSMIT

ReadProperty-Request,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Type

9. RECEIVE

ReadProperty-ACK,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

1. 帧过长

目的：验证接收帧状态机能检查对于IUT太大的帧的错误，该测试要验证如下接收帧状态机的转换：

HEADER\_CRC: FrameTooLong

BACnet参考条款:9.5.4.4.

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

Test\_Request,

'Length' = (x > 501),

'Data' = (x number of octets)

6. WAIT **Treply\_delay**

7. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Type,

8. RECEIVE

ReadProperty-ACK,

'Frame Type' = BACnet Data Not Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

1. 非法广播帧

目的：验证Slave Node有限状态机能拒绝一个非法广播帧。该测试要验证如下的Slave Node有限状态机的的转换：

IDLE: ReceivedUnwantedFrame (b)

BACnet参考条款：9.5.7.2.

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

DA = LOCAL BROADCAST,

'Frame Type' = BACnet Data Expecting Reply

(Any Confirmed service supported by IUT),

6. BEFORE (**Treply\_delay**) {

IF (receive a frame) THEN

ERROR "Response to broadcast confirmed service request."

}

7. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier

8. BEFORE (**Treply\_delay**) RECEIVE

ReadProperty-ACK,

'Frame Type' = BACnet Data Not Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier,

'Property Value' = (Device, X)

1. 非法广播测试请求帧

目的：验证Slave Node有限状态机能处理非法广播测试请求(Test\_Request)帧。该测试要验证如下的Slave Node有限状态机的转换：

IDLE: ReceivedUnwantedFrame (b)

BACnet参考条款：9.1.3 和 9.5.7.2.

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

DA = LOCAL BROADCAST,

Test\_Request

6. BEFORE (**Treply\_delay**) {

IF (receive a frame) THEN

ERROR "Response to broadcast Test\_Request."

}

7. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier

8. BEFORE (**Treply\_delay**) RECEIVE

ReadProperty-ACK,

'Frame Type' = BACnet Data Not Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier,

'Property Value' = (Device, X),

1. 不需要的令牌帧

目的：验证Slave Node有限状态机能拒绝不需要的令牌帧。

BACnet参考条款:9.5.7.2.

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

Token

6. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier

7. BEFORE (**Treply\_delay**) RECEIVE

ReadProperty-ACK,

'Frame Type' = BACnet Data Not Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier,

'Property Value' = (Device, X)

1. 不需要的主机轮询帧

目的：验证Slave Node有限状态机能拒绝不需要的主机轮询帧。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

Poll For Master

6. BEFORE (**Treply\_delay**) {

IF (receive a frame) THEN

ERROR "Slave device responded to Poll For Master."

}

7. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier

8. BEFORE (**Treply\_delay**) RECEIVE

ReadProperty-ACK,

'Frame Type' = BACnet Data Not Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier,

'Property Value' = (Device, X)

1. 不需要的主机轮询应答帧

目的：验证Slave Node有限状态机能拒绝不需要的主机轮询应答帧。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

Reply To Poll For Master

6. BEFORE (**Treply\_delay**) {

IF (receive a frame) THEN

ERROR "Slave device responded to Reply To Poll For Master."

}

7. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier

8. BEFORE (**Treply\_delay**) RECEIVE

ReadProperty-ACK,

'Frame Type' = BACnet Data Not Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier,

'Property Value' = (Device, X)

1. 不需要的延迟应答帧

目的：验证Slave Node有限状态机能拒绝不需要的延迟应答帧。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

Reply Postponed

6. BEFORE (**Treply\_delay**) {

IF (receive a frame) THEN

ERROR "Slave device responded to Reply Postponed."

}

7. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier

8. BEFORE (**Treply\_delay**) RECEIVE

ReadProperty-ACK,

'Frame Type' = BACnet Data Not Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier,

'Property Value' = (Device, X)

1. 应答测试

该条款定义了数据链错误被纠正或发布数据链错误应答的测试用例。

1. 重复前导码1

目的：验证接收帧状态机能检查重复的第一个前导码字节。该测试要验证如下接收帧状态机的转换：

PREAMBLE: RepeatedPreamble1

BACnet参考条款:9.5.4.2.

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request

4. RECEIVE

Test\_Response

5. TRANSMIT

X'55'

6. TRANSMIT

ReadProperty-Request,

'Frame Type' = BACnet Data Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier

7. BEFORE (**Treply\_delay**) RECEIVE

ReadProperty-ACK,

'Frame Type' = BACnet Data Not Expecting Reply,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier,

'Property Value' = (Device, X)

1. 测试请求帧

目的：验证可通过Test\_Response应答接收空白Test\_Request帧。该测试验证Slave Node有限状态机的以下转换：

IDLE: ReceivedDataNeedingReply

ANSWER DATA REQUEST: Reply

以及接收帧状态机的以下转换：

HEADER CRC: NoData

BACnet相关条款9.3.4.

测试步骤：在第4步应答中，帧间时隙应不大于**Tframe\_gap**.

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request,

'Length' = 0

4. BEFORE (**Treply\_delay**) RECEIVE

Test\_Response,

'Length' = 0

1. 带数据测试请求

目的：验证可通过Test\_Response应答接收携带数据的测试请求(Test\_Request)帧。

配置要求：第3步传输的Test\_Request帧的长度应小于或等于（IUT设备对象的 Max\_APDU\_Length\_Accepted 属性值）+ 21。在第4步应答中，帧间时隙应不大于**Tframe\_gap**。

测试步骤：

1. MAKE (IUT turned off or otherwise reset)

2. MAKE (IUT turned on or otherwise started)

3. TRANSMIT

Test\_Request,

'Length' = (x > 0),

'Data' = (x number of octets)

4. BEFORE (**Treply\_delay**) RECEIVE

Test\_Response

* 1. PTP 状态机测试

该节中定义的测试用例用于验证BACnet PTP半路由器可以进行在BACnet条款10中规定的通信。

条款13.2.1定义了一些测试。这些测试提供了一种简便方式表示在测试用例中多次出现的一系列测试步骤，并用符号方便地表示这些测试。每一个测试也用于验证特定的状态机的转换。在执行另一个测试时，13.2.1中定义的某一个测试失败则导致该测试失败。

在多种波特率下都可实现PTP协议。该条款中的所有测试应在IUT和DT支持的最高波特率下执行。验证IUT也可以支持其他波特率，在本条款中定义了可供测试人员选择的测试子集，供在IUT和DT均支持的每一个波特率下执行测试。

因为这是点对点（Point-to-Point，PTP）协议，以下测试的描述含有源端和/或终端的意思。在大多数测试用例中，字面上的发送端，意味着有其接收端。除非另有说明，帧的发送意味着无符号间的时间冲突的有效帧发送。

因为这也是个端对端 (peer-to-peer，PTP)协议，只要IUT处于CONNECTED状态，TD必须相应地行动，TD类似于在CONNECTED状态的PTP半路由器。除非在特定测试中另作说明，TD必须每**Theartbeat**发送一Heartbeat XON帧，并且必须以Data 0 Ack帧对Data 0帧应答，以Data 1 Ack帧对Data 1帧应答，以Test\_Response帧对Test\_Request帧应答，这些都要在**Tresponse**之内，并按BACnet条款10描述的格式进行。

因为测试TD的一致性不是这组测试的目的，TD本身不需要严格地实现BACnet条款10的所有功能。对维持IUT处于一种状态，从而可进行IUT一致性验证的进一步测试，TD上述应答行为就已足够了。

特别地，TD可能会选择不实施处理IUT发送的XON/XOFF帧。因为不要求IUT进行本身内部流控制决策的外部控制，IUT的流控制是不可检测的。但TD向IUT发送的流控制都要检测，这个规范包括了全面验证IUT一致性的测试。期望IUT在不向TD发送XOFF帧的情况下，能处理节奏适当的帧流。TD发送的帧流节奏可以在TD中通过配置参数设置，以防在正常测试中TD的帧流节奏超出IUT的接收能力。

为了便于观察，本条款中包含的测试步骤没有考虑IUT向TD发送XOFF帧的情况。这个规范并未排除TD处理传入流控制，但在TD里它的实现是可选择的，本条款未指定。如果TD选择实现对来自IUT的流控制处理，则由TD实现来决定在特定测试步骤的哪里IUT发送流控制比较合适，这可使用BACnet条款10中规定的操作。

部分测试需要TD发送期待应答，且包含NPDU的帧。这些用于验证DL-UNITDATA.indication和DL-UNITDATA.response的发送情况，并影响来自IUT的帧的传输。NPDU的选择由TD进行。如果供应商为TD提供了合理的应答时间上限，来代替这些测试中规定的**Tout**（参见BACnet 5.4.1），则许多测试的执行时间可极大地缩减。

1. PTP 测试
2. CONNECT\_TEST连接测试

用CONNECT\_TEST验证IUT通过最普通的路径可以由DISCONNECTED状态转换到CONNECTED状态。该测试将用于获得已知状态，从该状态开始进一步的测试步骤。因此，该测试大多直接执行到CONNECTED状态，无需其他可能状态转换的测试。

所需测试：无。

BACnet参考条款：10.4.8, 10.4.9, 10.4.9.1和10.4.9.3.

测试思想：验证以下的状态机，状态，和转换。

点对点连接状态机

DISCONNECTED: ConnectInbound (sending Connect Request)

INBOUND: ValidConnectResponseReceived (acceptance of no password or a correct password;

Transition to CONNECTED state is verified through its effect on Reception)

点对点传输状态机

TRANSMIT IDLE: ConnectionEstablishedXON

TRANSMIT BLOCKED or TRANSMIT READY: Heartbeat Timer Expired XON

点对点接收状态机

RECEIVE IDLE: ConnectionEstablished

连接状态机的初始状态： DISCONNECTED

连接状态机的终止状态： CONNECTED

测试步骤：

1. TRANSMIT "BACnet<CR>"

2. BEFORE **Tconn\_rqst**

RECEIVE Connect Request

3. TRANSMIT Connect Response,

'Password' = (any valid password)

4. TRANSMIT Heartbeat XON

5. BEFORE **Theartbeat** RECEIVE Heartbeat XON

注意事项：如果IUT不支持密码保护，省略第3步中的密码。

1. VERIFY\_CONNECTED\_TEST 验证连接测试

用VERIFY\_CONNECTED\_TEST验证IUT处于CONNECTED状态。用一个Test Request帧实现这个任务。该测试不会验证返回至RECEIVE READY状态的最后的转换，因此Test Request帧的测试不够充分。Test Request帧在12.2.4.2中陈述。

所需测试：无

BACnet参考条款:10.4.11.2.

测试思想：验证以下的状态机，状态和转换。

点对点接收状态机

RECEIVE READY: TestRequest (sending Test\_Response)

连接状态机的初始状态: CONNECTED

连接状态机的终止状态: CONNECTED

测试步骤：

1. TRANSMIT Disconnect Request

2. BEFORE **Tresponse** RECEIVE Test\_Response

注意事项: 在Test\_Request帧或Test\_Response帧中不能传输任何数据。

1. DISCONNECT\_TEST 断开连接测试

用DISCONNECT\_TEST验证IUT通过普通的路径可以从CONNECTED状态转换到DISCONNECTED状态。该测试将用于获得已知状态，从该状态开始进一步的测试步骤。因此，该测试大多直接执行到DISCONNECTED状态，无需其他可能状态转换的测试。

所需测试：无

BACnet参考条款:10.4.8, 10.4.9,和10.4.9.4.

测试思想：验证以下状态机，状态和转换。

点对点连接状态机

CONNECTED: DisconnectRequestReceived (sending Disconnect Response)

连接状态机的初始状态： CONNECTED

连接状态机的终止状态： DISCONNECTED

测试步骤：

1. TRANSMIT Disconnect Request

2. BEFORE **Tresponse** RECEIVE Disconnect Response

1. VERIFY\_DISCONNECTED\_TEST 验证断开连接测试

用VERIFY\_DISCONNECTED\_TEST验证IUT处于DISCONNECTED状态。用终止连接实现这个任务。该测试不会验证返回至DISCONNECTED状态的最后的转换，因此终止连接的测试不充足。

所需测试：无

BACnet参考条款:10.4.9.1和10.4.9.3.

测试思想：验证以下的状态机，状态和转换。

点对点连接状态机

DISCONNECTED: ConnectInbound (sending Connect Request)

INBOUND: DisconnectRequestReceived(sending Disconnect Response)

连接状态机的初始状态： DISCONNECTED

连接状态机的终止状态： DISCONNECTED

测试步骤：

1. TRANSMIT "BACnet<CR>"

2. BEFORE **Tconn\_rqst** RECEIVE Connect Request

3. TRANSMIT Disconnect Request

4. BEFORE **Tresponse** RECEIVE Disconnect Response

1. 建立连接

本条款定义的测试用来验证数据链路连接建立过程的正确实现以及在点对点连接状态机中正确地从DICONNECTED状态转换到CONNECTED状态。当初始PTP连接时，一个设备必须主动初始连接,而另一个设备则被动地等待连接的初始。

条款13.2.2.1中的测试适用于呼入连接,这时IUT被动地等待TD发送的触发字符串。条款13.2.2.2中的测试适用于呼出连接,这时IUT主动地向TD发送触发字符串。

1. 呼入连接测试

配置要求: 对于呼入连接测试, IUT必须是被动连接角色。IUT被配置为DISCONNECTED状态, 且在这些测试期间不得进入呼出（OUTBOUND）状态。在该测试期间，IUT不能初始断开连接序列。供应商应提供任何所需要的密码。

1. 呼入正常连接和断开连接测试

所需测试: 无。

BACnet参考条款: 10.4.8、10.4.9、10.4.9.1 和10.4.9.3。

目的:验证IUT能从DISCONNECTED状态转换到CONNECTED状态,并通过最正常的路径返回到DISCONNECTED状态。

测试思想: 实现 12.2.1中定义的测试,并验证与这些测试有关的状态转换。

连接状态机的初始状态: DISCONNECTED

连接状态机的终止状态: DISCONNECTED

测试步骤：

1. CONNECT\_TEST

2. VERIFY\_CONNECTED\_TEST

3. DISCONNECT\_TEST

4. VERIFY\_DISCONNECTED\_TEST

1. 重试失败的呼入连接测试

所需测试：无。

BACnet参考条款:10.4.9.1和10.4.9.3。

目的:验证IUT在建立连接期间能适当地使用重试和超时。此测试结果为连接失败。

测试思想:验证以下状态机、状态和转换:

点对点连接状态机

DISCONNECTED: ConnectInbound (sending Connect Request; RetryCount and ResponseTimer settings)

INBOUND: ConnectResponseTimeout (ResponseTimer expiration ; resending Connect Request)

INBOUND: ConnectResponseFailure (RetryCount exhaustion; transition to DISCONNECTED state)

连接状态机的初始状态: DISCONNECTED

连接状态机的终止状态: DISCONNECTED

测试步骤：

1. TRANSMIT "BACnet<CR>"

2. BEFORE **Tconn\_rqst** RECEIVE Connect Request

3. WHILE (RetryCount < Nretries) DO {

WAIT **Tconn\_rqst**

RECEIVE Connect Request

}

4. VERIFY\_DISCONNECTED\_TEST

测试人员注意：WHILE循环表示给IUT配置的重试次数。在重试次数用完之前要不断地重试连接。

1. 重试成功的呼入连接测试

所需测试：ReadProperty服务执行测试, 9.18。

BACnet参考条款:10.4.9.1和10.4.9.3。

目的: 验证IUT在建立连接期间能正确使用重试和超时,并在初始连接失败后实现连接。

测试思想:验证以下状态机、状态和转换:

点对点连接状态机

DISCONNECTED: ConnectInbound (RetryCount and ResponseTimer settings)

INBOUND: ConnectResponseTimeout (ResponseTimer expiration; resending of Connect Request)

连接状态机的初始状态: DISCONNECTED

连接状态机的终止状态: CONNECTED

测试步骤：

1. TRANSMIT "BACnet<CR>"

2. BEFORE **Tconn\_rqst** RECEIVE Connect Request

3. WAIT **Tconn\_rqst**

4. RECEIVE Connect Request

5. TRANSMIT Connect Response,

'Password' = (any valid password)

6. TRANSMIT Heartbeat XON

7. BEFORE **Theartbeat** RECEIVE Heartbeat XON

8. VERIFY (the Device object of the IUT), Object\_Name = (the value specified in the EPICS)

9. VERIFY\_CONNECTED\_TEST

1. 终止呼入连接测试

所需测试：无。

BACnet参考条款:10.4.9.1和10.4.9.3。

目的: 验证IUT能正确终止连接进程。该测试过程要运行两次,以验证从INBOUND状态到DISCONNECTED状态的转换。

测试思想:验证以下状态机、状态和转换:

点对点连接状态机

DISCONNECTED: ConnectInbound (sending Connect Request)

INBOUND: DisconnectRequestReceived (sending Disconnect Response; transition to DISCONNECTED)

连接状态机的初始状态: DISCONNECTED

连接状态机的终止状态: DISCONNECTED

测试步骤：

1. VERIFY\_DISCONNECTED\_TEST

2. VERIFY\_DISCONNECTED\_TEST

1. 重新连接测试

所需测试：无。

BACnet参考条款:10.4.9.4。

目的: 验证IUT在连接状态下能正确应答连接请求。

测试思想: 本测试用例验证以下状态机、状态和转换:

点对点连接状态机

CONNECTED: ConnectRequestReceived (sending Connect Response)

连接状态机的初始状态: CONNECTED

连接状态机的终止状态: CONNECTED

测试步骤：

1. CONNECT\_TEST

2. TRANSMIT Connection Request

3. BEFORE **Tresponse** RECEIVE Connect Response

4.VERIFY\_CONNECTION\_TEST

1. 呼出连接测试

配置要求: 对于呼出连接测试，IUT必须主动地执行连接。应将IUT配置为以DISCONNECTED状态开始,并在可能的情况下保持DISCONNECTED状态,直到特定的触发条件从而执行连接。

1. 呼出连接正常测试

所需测试：无。

BACnet参考条款: 10.4.9.1, 10.4.9.2和10.4.9.4。

目的: 验证IUT能正确开始并成功地建立呼出连接。

测试思想: 本测试用例验证以下状态机、状态和转换:

点对点连接状态机

DISCONNECTED： ConnectOutbound（receipt of the DL-CONNECT.request is implied;

sending trigger sequence）

OUTBOUND： ConnectRequestReceived（sending Connect Response）

连接状态机的初始状态: DISCONNECTED

连接状态机的终止状态: CONNECTED

测试步骤：

1. MAKE (the IUT initiate a connection sequence)

2. RECEIVE "BACnet<CR>"

3. TRANSMIT Connect Request

4. BEFORE **Tconn\_rsp** RECEIVE Connect Response

5. TRANSMIT Heartbeat XON

6. BEFORE **Theartbeat** RECEIVE Heartbeat XON

7. VERIFY (the Device object of the IUT), Object\_Name = (the value specified in the EPICS)

8. VERIFY\_CONNECTED\_TEST

1. 重试成功的按需连接测试

所需测试：无。

BACnet参考条款: 10.4.9.1, 10.4.9.2和10.4.9.4。

目的: 验证尝试发起呼出连接失败时，IUT是否重试。

测试思想: 本测试用例验证以下状态机、状态和转换:

点对点连接状态机

DISCONNECTED：ConnectOutbound（receipt of the DL-CONNECT.request;

sending trigger sequence;RetryCount and ResponseTimer settings）

OUTBOUND： ConnectRequestTimeout（resending trigger sequence; ResponseTimer setting）

OUTBOUND： ConnectRequestReceived（sending Connect Response）

连接状态机的初始状态: DISCONNECTED

连接状态机的终止状态: CONNECTED

测试步骤：

1. MAKE (the IUT initiate a connection sequence)

2. RECEIVE "BACnet<CR>"

3. WAIT **Tconn\_rsp**

4. RECEIVE "BACnet<CR>"

5. TRANSMIT Connect Request

6. RECEIVE Connect Response

7. TRANSMIT Heartbeat XON

8. BEFORE **Theartbeat** RECEIVE Heartbeat XON

9. VERIFY (the Device object of the IUT), Object\_Name = (the value specified in the EPICS)

10. VERIFY\_CONNECTED\_TEST

1. 重试失败的按需连接测试

所需测试：无。

BACnet参考条款: 10.4.9.1, 10.4.9.2和10.4.9.4。

目的: 验证在尝试发起呼出连接失败时IUT是否重试，并能在重试次数超过**Nretries**后停止重试。

测试思想: 本测试用例验证以下状态机、状态和转换:

点对点连接状态机

DISCONNECTED: ConnectOutbound (receipt of the DL-CONNECT.request;

sending trigger sequence;RetryCount and ResponseTimer settings)

OUTBOUND: ConnectRequestTimeout (resending trigger sequence; ResponseTimer setting)

OUTBOUND: ConnectRequestFailure

连接状态机的初始状态: DISCONNECTED

连接状态机的终止状态: DISCONNECTED

测试步骤：

1. MAKE (the IUT initiate a connection sequence)

2. RECEIVE "BACnet<CR>"

3. WHILE (RetryCount < Nretries) DO {

WAIT **Tconn\_rsp**

RECEIVE "BACnet<CR>"

}

4. VERIFY\_DISCONNECTED\_TEST

1. 连接终止

本条款定义的测试验证在点对点连接状态机中从CONNECTED状态到DISCONNECTED状态的的所有路径。对于给定的IUT, 某些路径可能无法测试。

1. 正常网络断开测试

所需测试：无。

BACnet参考条款: 10.4.9.1, 10.4.9.3和10.4.9.5。

目的: 验证IUT能断开已建立的PTP连接。

测试思想: 本测试用例验证以下状态机、状态和转换:

点对点连接状态机

CONNECTED: NetworkDisconnect (sending the Disconnect Request frame)

DISCONNECTING: DisconnectResponseReceived

连接状态机的初始状态: CONNECTED

连接状态机的终止状态: DISCONNECTED

配置要求: 在测试开始之前, 应在IUT和TD之间建立有效的PTP连接。

测试步骤：

1. MAKE (the IUT initiate a disconnect)

2. RECEIVE Disconnect Request

3. TRANSMIT Disconnect Response

4. WAIT **Tresponse**

5. VERIFY\_DISCONNECT\_TEST

1. 重试网络断开测试

所需测试：无。

BACnet参考条款: 10.4.9.1, 10.4.9.3，10.4.9.4和10.4.9.5。

目的: 验证IUT在发送断开连接请求后，即使未收到断开连接应答,最终还是能转换为DISCONNECTED状态。若不能使IUT发出断开连接请求,则此测试可以省略。

测试思想: 本测试用例验证以下状态机、状态和转换:

点对点连接状态机

DISCONNECTED: ConnectInbound (sending Connection Request frame)

INBOUND: ConnectResponseTimeout

DISCONNECTING: DisconnectResponseTimeout (ResponseTimer and RetryCount settings)

DISCONNECTING: DisconnectResponseFailure (correct RetryCount settings when leaving INBOUND state)

CONNECTED: NetworkDisconnect (sending Disconnect Request frame; ResponseTimer and RetryCount settings)

连接状态机的初始状态: CONNECTED

连接状态机的终止状态: DISCONNECTED

配置要求: 在测试开始之前, 应在IUT和TD之间建立有效的PTP连接。在此测试期间, IUT处于DISCONNECTED状态时,应设置为不发起连接。

测试步骤：

1. MAKE (the IUT initiate a disconnect)

2. RECEIVE Disconnect Request

3. WHILE (RetryCount < **Nretries**) DO {

WAIT **Tresponse**

RECEIVE Disconnect Request

}

4. WAIT (**Tresponse** + 5 seconds)

5. VERIFY\_DISCONNECT\_TEST

1. 不需要的帧不断开连接测试

所需测试：无。

BACnet参考条款:10.4.9.5。

目的: 验证IUT在DISCONNECTING状态时接收到不需要的帧不会影应答计时器。

测试思想: 有两种方法可以将IUT置于DISCONNECTING状态:在连接时提供无效的密码,或者一种本地方法使IUT初始断开连接序列。下面的测试步骤对上述每种情况都进行一个分支测试，其中任何一种方法都可以使用。一旦IUT进入断开连接状态, 此测试用例将验证以下状态机、状态和转换:

点对点连接状态机

DISCONNECTING： UnwantedFrameReceived（no affect on ResponseTimer）

连接状态机的初始状态: DISCONNECTED 或者 CONNECTED

连接状态机的终止状态: DISCONNECTED

配置要求: 如果使用无效密码的方法强制IUT进入DISCONNECTING状态, 则应将IUT设置为需要密码, 并以DISCONNECTED状态开始测试。如果使用IUT发起断开连接序列的方法, 则应在测试开始前在IUT和TD之间建立有效的PTP连接。

测试步骤：

1. IF (the invalid password method it to be used for causing the IUT to enter the DISCONNECTING state) THEN

TRANSMIT "BACnet<CR>"

BEFORE **Tconn\_rqst** RECEIVE Connect Request

TRANSMIT Connect Response

Password = (missing or invalid password)

ELSE

MAKE (the IUT initiate a disconnect)

2. BEFORE **Tresponse** RECEIVE Disconnect Request

3. WAIT (**Tresponse**/2)

4. TRANSMIT Connect Request

5. BEFORE **Tresponse** RECEIVE Disconnect Request

6. TRANSMIT Disconnect Response

7. VERIFY\_DISCONNECT\_TEST

1. 同时断开连接测试

所需测试：无。

BACnet参考条款:10.4.9.5。

目的: 验证从DISCONNECTING状态到DisconnectRequestReceived转换。

测试思想: 有两种方法可以将IUT置于DISCONNECTING状态:在连接时提供无效的密码, 或者用一种本地方法使IUT初始断开连接序列。下面的测试步骤对上述每种情况都进行一个分支测试，可使用其中任一种方法。一旦IUT进入DISCONNECTING状态,此测试验证以下状态机、状态和转换:

点对点连接状态机

DISCONNECTING ： DisconnectRequestReceived（Sending Disconnect Response frame）

连接状态机的初始状态: DISCONNECTING

连接状态机的终止状态: DISCONNECTED

配置要求: 如果使用无效密码的方法强制IUT进入DISCONNECTING状态,则应将IUT设置为需要密码,并以DISCONNECTED状态开始测试。如果使用IUT将初始断开连接序列的方法,则应在测试开始前在IUT和TD之间建立有效的PTP连接。

测试步骤：

1. IF (the invalid password method it to be used for causing the IUT to enter the DISCONNECTING state) THEN

TRANSMIT "BACnet<CR>"

BEFORE **Tconn\_rqst** RECEIVE Connect Request

TRANSMIT Connect Response

Password = (missing or invalid password)

ELSE

MAKE (the IUT initiate a disconnect)

2. BEFORE **Tresponse** RECEIVE Disconnect Request

3. TRANSMIT Disconnect Request

4. BEFORE **Tresponse** RECEIVE Disconnect Response

5. VERIFY\_DISCONNECT\_TEST

1. 无效密码的断开连接测试

所需测试：无。

BACnet参考条款: 10.4.9.1, 10.4.9.3,和10.4.9.5

目的: 验证由于密码无效，IUT是否能正确中止连接进程。如果IUT不能配置为需要密码,则不执行此测试。

测试思想: 此测试验证以下状态机、状态和转换:

点对点连接状态机

DISCONNECTED： ConnectInbound（sending Connect Request frame）

INBOUND： InvalidConnectResponseReceived（sending Disconnect Request frame）

DISCONNECTING： DisconnectResponseReceived

连接状态机的初始状态: DISCONNECTED

连接状态机的终止状态: DISCONNECTED

配置要求:IUT应配置为需要密码。

测试步骤：

1. TRANSMIT "BACnet<CR>"

2. BEFORE **Tconn\_rqst** RECEIVE Connect Request

3. TRANSMIT Connect Response

“Password” = (an invalid password)

4. BEFORE **Tresponse** RECEIVE Disconnect Request

5. TRANSMIT Disconnect Response

6. VERIFY\_DISCONNECT\_TEST

1. 未提供密码的断开连接测试

所需测试：无。

BACnet参考条款: 10.4.9.1, 10.4.9.3,和10.4.9.5

目的: 验证由于未提供密码，IUT是否能正确中止连接进程。如果IUT不能配置为需要密码, 则不执行此测试。

测试思想: 此测试用例验证以下状态机、状态和转换:

点对点连接状态机

DISCONNECTED： ConnectInbound（sending Connect Request frame）

INBOUND： InvalidConnectResponseReceived (sending Disconnect Request frame）

DISCONNECTING： DisconnectResponseReceived

连接状态机的初始状态: DISCONNECTED

连接状态机的终止状态: DISCONNECTED

配置要求: IUT应配置为需要密码。

测试步骤：

1. TRANSMIT "BACnet<CR>"

2. BEFORE **Tconn\_rqst** RECEIVE Connect Request

3. TRANSMIT Connect Response

'Password' = (no password shall be provided)

4. BEFORE **Tresponse** RECEIVE Disconnect Request

5. TRANSMIT Disconnect Response

6. VERIFY\_DISCONNECT\_TEST

1. 重试拒绝密码断开连接测试

所需测试：无。

BACnet参考条款: 10.4.9.3, 10.4.9.4,和10.4.9.5

目的: 验证IUT在应答连接建立过程中的无效密码时，能正确转换为DISCONNECTED状态。此测试还验证在从INBOUND状态到InvalidConnectResponseReceived转换上，应答计时器和重试计数器是否正确设置。如果IUT不能配置为支持密码,则不应执行此测试。

测试思想: TD发起连接序列,并强制IUT重试连接请求,以使重试计数器递增。然后, TD使用无效的密码发送连接应答,使IUT发起断开连接。TD不对断开连接请求应答，以使IUT不断地重试，直到到达重试上限。所发送的重试次数表示了重试计数器是否正确重置。此测试用例验证以下状态机、状态和转换:

点对点连接状态机

DISCONNECTED: ConnectInbound (sending Connection Request frame)

INBOUND: InvalidConnectResponseReceived (sending Disconnect Request frame; RetryCount

and ResponseTimer settings)

DISCONNECTING: DisconnectResponseTimeout (ResponseTimer and RetryCount settings)，

DisconnectResponseFailure (correct RetryCount settings when leaving INBOUND state)

连接状态机的初始状态: DISCONNECTED

连接状态机的终止状态: DISCONNECTED

配置要求:IUT应配置为需要密码才能建立连接,并且在测试期间处于断开连接状态时，不启动连接。

测试步骤：

1. TRANSMIT "BACnet<CR>"

2. BEFORE **Tconn\_rqst** RECEIVE Connect Request

3. WAIT **Tconn\_rsp**

4. RECEIVE Connect Request

5. TRANSMIT Connect Response

'Password' = (no password shall be provided)

6. BEFORE **Tresponse** RECEIVE Disconnect Request

7. WAIT **Tresponse**

8. RECEIVE Disconnect Request

9. WAIT **Tresponse**

10. RECEIVE Disconnect Request

11. WAIT **Tresponse**

12. RECEIVE Disconnect Request

13. WAIT **Tresponse**

14. VERIFY\_DISCONNECT\_TEST

1. 被动重连的物理连接丢失测试

所需测试：无。

BACnet参考条款:10.4.9.4

目的: 验证当物理连接断开时, IUT能正确返回到DISCONNECTED状态。如果IUT会自发地尝试重新连接,则此测试应省略。如果IUT不能检测到物理链路丢失,则此测试应忽略。

测试思想: 测试从已建立的PTP连接开始。TD断开物理连接,并等待IUT检测连接丢失。然后TD恢复连接,并验证TD是否处于DISCONNECTED状态。此测试用例验证以下状态机、状态和转换:

点对点连接状态机

CONNECTED： ConnectionLost

连接状态机的初始状态: CONNECTED

连接状态机的终止状态: DISCONNECTED

测试步骤：

1. MAKE (the TD disrupt the physical connection or simulate a disruption)

2. WAIT (5 seconds)

3. MAKE (the TD restore the physical connection)

4. VERIFY\_DISCONNECT\_TEST

1. 主动重连的物理连接丢失测试

所需测试：无。

BACnet参考条款:10.4.9.4.

目的: 验证当物理连接断开时, IUT能正确返回到DISCONNECTED状态。如果IUT不会自动地尝试重新连接,则此测试应省略。如果IUT无法检测到物理链路丢失,则此测试应省略。

测试思想: 测试从已建立的PTP连接开始。TD断开物理连接,并等待IUT检测连接丢失后，TD再恢复连接,并验证TD是否尝试重新建立连接。此测试用例验证以下状态机、状态和转换:

点对点连接状态机

CONNECTED： ConnectionLost

连接状态机的初始状态: CONNECTED

连接状态机的终止状态: DISCONNECTED

测试步骤：

1. MAKE (the TD disrupt the physical connection or simulate a disruption)

2. WAIT (5 seconds)

3. MAKE (the TD restore the physical connection)

4. RECEIVE "BACnet<CR>"

1. 不活动的断开连接测试

所需测试：无。

BACnet参考条款:10.4.9.4

目的: 验证IUT在连接状态下是否正确超时, 以及IUT是否能单方面进入断开连接状态。

测试思想: 测试从已建立的PTP连接开始。TD保持停止状态的时间足够长,使非活动的计时器过期。然后验证IUT是否已转换为DISCONNECTED状态。此测试用例验证以下状态机、状态和转换:

点对点连接状态机

CONNECTED： Inactivity Timeout

连接状态机的初始状态: CONNECTED

连接状态机的终止状态: DISCONNECTED

测试步骤：

1. WAIT (**Tinactivity** + 5 seconds)

2. VERIFY\_DISCONNECT\_TEST

1. 接收

本测试接收状态机规定的状态和转换。

接收状态机的某些状态和转换将在传输状态机测试中进行验证。其中包括DATA ACK状态和DATA NAK 状态中的所有转换，以及RECEIVE READY状态的DataAck、DataNak,Heartbeat XON和Heartbeat XOFF转换。

因为TD无法影响IUT的内部流控制，DATA状态外的Duplicate0\_FullBuffers, Duplicate1\_FullBuffers, Data0\_FullBuffers, Data1\_FullBuffers,LastData0和Lastdata1转换。RECEIVE READY状态的 BadData0\_FullBuffers, BadData1\_FullBuffers转换不可测试。

配置要求: 配置IUT为在建立连接后不会发起已确认请求。

1. 常接收序列测试

所需测试：无。

BACnet参考条款: 10.4.11.2 和 10.4.11.3.

目的: 验证IUT是否在建立连接时正确初始化了RxSequenceNumber，以及按正确的顺序接收数据帧。首先验证序列号能正确地从0转到1, 然后再返回。然后,当序列号为1时断开连接,以验证序列号的初始设置, 再连接从而验证序列号能返回0。本测试继续测试是否能正确处理重复数据帧。

测试思想: 由于IUT总是通过Data Ack0帧确认Data 0帧，通过Data Ack1帧确认Data 1帧, 而不考虑内部RxSequenceNumber。在不能验证是否发送了DL-UNITDATA.indication, 或者是否丢弃了该帧的情况下，RxSequenceNumber实现的实际一致性是无法验证的。因此TD仅使用数据链路层消息不能验证数据链路层的一致性。基于这个原因，此测试偶尔会使用ReadProperty服务请求。在每种情况下, 读取的属性都是IUT的设备对象的Object\_Type。这可确保不需要进行分段。此测试用例验证以下状态机、状态和转换:

点对点接收状态机

RECEIVE IDLE: ConnectionEstablished (setting RxSequenceNumber to 0)

RECEIVE READY: DataReceived

DATA: NewData0 (sending Data Ack 0 XON frame; setting RxSequenceNumber to 1)

NewData1 (sending Data Ack 1 XON frame; setting RxSequenceNumber to 0)

Duplicate0 (sending Data Ack 0 XON frame; discarding duplicate frame)

Duplicate1 (sending Data Ack 1 XON frame; discarding duplicate frame)

连接状态机的初始状态: DISCONNECTED

连接状态机的终止状态: DISCONNECTED

测试步骤：

1. CONNECT\_TEST

2. TRANSMIT Data 0,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

3. BEFORE **Tresponse** RECEIVE Data Ack 0 XON

4. BEFORE **Tout** RECEIVE Data 0 | Data 1,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

5. TRANSMIT Data 1,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

6. BEFORE **Tresponse** RECEIVE Data Ack 1 XON

7. BEFORE **Tout** RECEIVE Data 0 | Data 1,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

8. TRANSMIT Data 0,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

9. BEFORE **Tresponse** RECEIVE Data Ack 0 XON

10. BEFORE **Tout** RECEIVE Data 0 | Data 1,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

11. DISCONNECT\_TEST

12. CONNECT\_TEST

13. TRANSMIT Data 0,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

14. BEFORE **Tresponse** RECEIVE Data Ack 0 XON

15. BEFORE **Tout** RECEIVE Data 0 | Data 1,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

16. TRANSMIT Data 0,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

17. BEFORE **Tresponse** RECEIVE Data Ack 0 XON

18. WAIT (**Tout** + 5 seconds)

19. CHECK (verify that the IUT did not transmit a response)

20. TRANSMIT Data 1,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

21. BEFORE **Tresponse** RECEIVE Data Ack 1 XON

22. BEFORE **Tout** RECEIVE Data 0 | Data 1,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

23. TRANSMIT Data 1,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

24. BEFORE **Tresponse** RECEIVE Data Ack 1 XON

25. WAIT (**Tout** + 5 seconds)

26. CHECK (verify that the IUT did not transmit a response)

27. DISCONNECT\_TEST

注意事项:在步骤4、7、10、15和22中使用数据0或数据1进行应答不是关键。这取决于从IUT可能发送的之前的数据帧。

1. Test\_Request测试

所需测试：无。

BACnet参考条款: 10.4.11.2

目的: 此测试用例验证IUT能正确应答Test\_Request帧。验证接收Test\_Request帧不会影响RxSequenceNumber。BACnet不指定对Test\_Response的等待时间量。此测试假定时间**Tresponse**足够长。

测试思想: 建立一个新的连接, 以确保RxSequenceNumber为0。TD发送确认服务请求，使RxSequenceNumber翻转为1。TD发送一个传输数据的Test\_Request。IUT应使用Test\_Response应答。TD会发送另一个确认服务请求, 以验证RxSequenceNumber仍然为1。所使用的已确认服务请求是 ReadProperty请求。读取的属性是IUT设备目标的Object\_Type。这可确保不需要进行分段。此测试用例验证以下状态机、状态和转换:

点对点接收状态机

RECEIVE READY ： TestRequest（sending the Test\_Response frame; that is has no effect on

RxSequenceNumber）

连接状态机的初始状态: DISCONNECTED

连接状态机的终止状态: DISCONNECTED

测试步骤：

1. CONNECT\_TEST

2. TRANSMIT Data 0,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

3. BEFORE **Tresponse** RECEIVE Data Ack 0 XON

4. BEFORE **Tout** RECEIVE Data 0 | Data 1,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

5. TRANSMIT Test\_Request,

'Data' = (any test data selected by the TD)

6. BEFORE **Tresponse** RECEIVE Test\_Response,

'Data' = (the data transmitted in step 5)

7. TRANSMIT Data 1,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

8. BEFORE **Tresponse** RECEIVE Data Ack 1 XON

9. BEFORE **Tout** RECEIVE Data 0 | Data 1,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

10. DISCONNECT\_TEST

1. 重新连接接收序列测试

所需测试：无。

BACnet参考条款: 10.4.9.4

目的: 验证IUT在从点对点连接状态机的CONNECTED状态进行ConnectRequestReceived转换时，是否重置序列。如从INBOUND状态到ValidConnectResponseReceived转换一样，此转换发出DL-CONNECT.indication，但RxSequenceNumber和TxSequenceNumber均不应重置。

测试思想:此测试用例验证以下状态机、状态和转换:

点对点连接状态机

CONNCETED： ConnectRequestReceived（sending Connect Response frame; that it has no effect

on RxSequenceNumber）

连接状态机的初始状态: DISCONNECTED

连接状态机的终止状态: DISCONNECTED

测试步骤：

1. CONNECT\_TEST

2. TRANSMIT Data 0,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

3. BEFORE **Tresponse** RECEIVE Data Ack 0 XON

4. TRANSMIT Connect Request

5. BEFORE **Tconn\_rsp** RECEIVE Connect Response

6. TRANSMIT Data 1,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

7. BEFORE **Tresponse** RECEIVE Data Ack 1 XON

8. BEFORE **Tout** RECEIVE Data 0 | Data 1,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

9. DISCONNECT\_TEST

1. 错误数据测试

所需测试：无。

BACnet参考条款: 10.4.11.2

目的: 此测试用例验证IUT是否发送正确的Data Nak帧以应答错误数据帧。测试还验证错误帧对RxSequenceNumber没有影响。

测试思想: 建立一个新的连接, 确保RxSequenceNumber为0。TD发送一个包含确认请求和错误CRC的Data 1帧。IUT应该用NAK应答并丢弃消息。TD发送一个包含确认请求的正确的Data 0帧，来验证RxSequenceNumber是否未更改。TD发送另一个有错误CRC的Data 1帧，后跟一个正确的Data 1帧,用来验证序列1的相同行为。此测试用例验证以下状态机、状态和转换:

点对点接收状态机

RECEIVE READY：BadData0（sending Data Nak 0 XON frame; discarding the frame; no effect on RxSequenceNumber）

BadData1（sending Data Nak 1 XON frame; discarding the frame; no effect on RxSequenceNumber）

连接状态机的初始状态: DISCONNECTED

连接状态机的终止状态: DISCONNECTED

测试步骤：

1. CONNECT\_TEST

2. TRANSMIT Data 1,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

CRC = (any bad or missing value)

3. BEFORE **Tresponse** RECEIVE Data Nak 1 XON

4. WAIT **Tout**

5. MAKE (verify that the IUT did not respond to the ReadProperty request)

6. TRANSMIT Data 0,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

7. BEFORE **Tresponse** RECEIVE Data Ack 0 XON

8. BEFORE **Tout** RECEIVE Data 0 | Data 1,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

9. TRANSMIT Data 1,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

CRC = (any bad or missing value)

10. BEFORE **Tresponse** RECEIVE Data Nak 1 XON

11. WAIT **Tout**

12. MAKE (verify that the IUT did not respond to the ReadProperty request)

13. TRANSMIT Data 0,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

14. BEFORE **Tresponse** RECEIVE Data Ack 0 XON

15. BEFORE **Tout** RECEIVE Data 0 | Data 1,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

16. DISCONNECT\_TEST

1. 重复确认测试

所需测试：无。

BACnet参考条款: 10.4.11.4

目的: 该测试通过验证重复的ack对传输重试没有影响来验证它们是否被丢弃。它还验证了重复确认的XON/XOFF方面。

测试思想: TD向IUT发送已确认请求。当IUT应答时, TD发送一个Nak XON来强制重试。当第一次重试被接收到时，TD发送重复的Ack XON，并验证是否在适当的时间收到第二次重试,且不受收到重复ACK的影响。TD发送一个重复的Ack XOFF，验证第三次重试是否在正确的时间到达,且不受到重复ACK的影响。之后测试XOFF的效果。TD发送一个Ack XON,并验证XON是否被正确译码。重复该过程以测试相对的序列号。此测试用例验证以下状态机、状态和转换:

点对点接收状态机

DATA ACK： Duplicate\_XON（no effect on TRANSMIT PENDING state; proper

TransmissionBlockled setting）

Duplicate\_XOFF:（no effect on TRANSMIT PENDING state; proper

TransmissionBlockled setting）

连接状态机的初始状态: CONNECTED

连接状态机的终止状态: CONNECTED

REPEAT X = (data sequence number 0 and 1) DO {

1. TRANSMIT Data 1,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

2. BEFORE **Tout** RECEIVE Data 0 | Data 1

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

3. IF (a Data 0 frame was received in step 3) THEN

TRANSMIT Data Nak 0 XON

ELSE

TRANSMIT Data Nak 1 XON

4. BEFORE **Tout** RECEIVE Data 0 | Data 1

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

5. WAIT (**Tresponse** /2)

6. IF (a Data 0 frame was received in step 5) THEN

TRANSMIT Data Ack 1 XON

ELSE

TRANSMIT Data Ack 0 XON

7. BEFORE (**Tresponse**/2) RECEIVE Data 0 | Data 1

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

8. WAIT (**Tresponse** /2)

9. IF (a Data 0 frame was received in step 8) THEN

TRANSMIT Data Ack 1 XOFF

ELSE

TRANSMIT Data Ack 0 XOFF

10. BEFORE (**Tresponse**/2) RECEIVE Data 0 | Data 1

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

11. WAIT (**Tresponse** + 2 seconds)

12. CHECK (verify that no additional retries have been transmitted)

13. TRANSMIT Data 1,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

14. WAIT (**Tresponse** + 2 seconds)

15. CHECK (verify that no response has been transmitted)

16. IF (a Data 0 frame was received in step 8) THEN

TRANSMIT Data Ack 1 XON

ELSE

TRANSMIT Data Ack 0 XON

17. BEFORE **Tout** RECEIVE Data 0 | Data 1

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

}

1. 重复的否定应答测试

所需测试：无。

BACnet参考条款: 10.4.11.5

目的：该测试通过验证重复否定应答对传输重试没有影响来验证是否丢弃了重复否定应答。它还可以验证重复否定应答的XON/XOFF方面。

测试思想: TD向IUT发送已确认请求。当IUT应答时，TD发送一个NAK XON来强制重试。当收到第一次重试时，TD发送一个重复的NAK XON，并验证第二次重试是否在正确的时间收到，且不受收到重复否定应答的影响。TD发送一个重复的Nak XOFF，并验证第三次重试是否在正确的时间到达，且不受收到重复否定应答的影响。然后测试XOFF的效果。TD发送一个Nak XON, 然后验证XON是否被正确译码。重复该过程以测试相对的序列号。此测试用例验证以下状态机、状态和转换:

点对点接收状态机

DATA NAK： Duplicate\_XON（no effect on TRANSMIT PENDING state; proper TransmissionBlocked

setting）

Duplicate\_XOFF（no effect on TRANSMIT PENDING state; proper TransmissionBlocked

setting）

连接状态机的初始状态: CONNECTED

连接状态机的终止状态: CONNECTED

REPEAT X = (data sequence number 0 and 1) DO {

1. TRANSMIT Data 1,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

2. BEFORE **Tout** RECEIVE Data 0 | Data 1,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

3. IF (a Data 0 frame was received in step 3) THEN

TRANSMIT Data Nak 0 XON

ELSE

TRANSMIT Data Nak 1 XON

4. BEFORE **Tout** RECEIVE Data 0 | Data 1,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

5. WAIT (**Tresponse** /2)

6. IF (a Data 0 frame was received in step 4) THEN

TRANSMIT Data Nak 1 XON

ELSE

TRANSMIT Data Nak 0 XON

7. BEFORE (**Tresponse** /2) RECEIVE Data 0 | Data 1,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

8. WAIT (**Tresponse** /2)

9. IF (a Data 0 frame was received in step 7) THEN

TRANSMIT Data Nak 1 XOFF

ELSE

TRANSMIT Data Nak 0 XOFF

10. BEFORE (**Tresponse** /2) RECEIVE Data 0 | Data 1,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

11. WAIT (**Tresponse** + 2 seconds)

12. CHECK (verify that no additional retries have been transmitted)

13. TRANSMIT Data 1,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

14. WAIT (**Tresponse** + 2 seconds)

15. CHECK (verify that no response has been transmitted)

16. IF (a Data 0 frame was received in step 7) THEN

TRANSMIT Data Nak 1 XON

ELSE

TRANSMIT Data Nak 0 XON

17. BEFORE **Tout** RECEIVE Data 0 | Data 1,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

}

1. 传输

本条款测试传输状态机规定的状态和转换，也对接收状态机的一些状态和转换进行测试。

配置要求: 应配置IUT为在建立连接后不会发起确认请求。

1. 发起传输连接和断开测试

所需测试：无。

BACnet参考条款: 10.4.10.1和10.4.10.2

目的: 此测试用例验证IUT能正确地从TRANSMIT IDLE状态转换为TRANSMIT BLOCKED状态并返回。本测试要执行两次以验证最终转换回TRANSMIT IDLE状态。

测试思想：遵循正常的连接过程，只是IUT保留heartbeat帧以强制其保持在TRANSMIT BLOCKED状态。此测试用例验证以下状态机、状态和转换:

点对点传输状态机

TRANSMIT IDLE： ConnectionEstablishedXON（sending the Heartbeat XON frame）

TRANSMIT BLOCKED：HeartbeatTimerExpiredXON（sending the Heartbeat XON frame）

SendRequest

Disconnected

连接状态机的初始状态: DISCONNECTED

连接状态机的终止状态: DISCONNECTED

测试步骤：

1. TRANSMIT "BACnet<CR>"

2. BEFORE **Tconn\_rqst**

RECEIVE Connect Request

3. TRANSMIT Connect Response,

'Password' = (any valid password)

4. BEFORE **Theartbeat** RECEIVE Heartbeat XON

5. TRANSMIT Data 0,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

6. REPEAT X = (1, 2, 3, and 4) DO {

WAIT **Theartbeat**

TRANSMIT Heartbeat XOFF

}

7. CHECK (verify that the IUT has not responded to the read request)

8. DISCONNECT\_TEST

9. CONNECT\_TEST

10. DISCONNECT\_TEST

1. 发送准备就绪测试

所需测试：无。

BACnet参考条款: 10.4.10.3

目的:此测试用例验证正确设置HeartbeatTimer以及从TRANSMIT READY状态到Disconnected的转换。HeartbeatTimerExpiredXOFF是不可测试的。

测试思想：此测试用例验证以下状态机、状态和转换:

点对点传输状态机

TRANSMIT READY：HeartbeatTimerExpiredXON（sending the Heartbeat XON frame; setting the

HeartbeatTimer）

TRANSMIT READY：Disconnected

连接状态机的初始状态: DISCONNECTED

连接状态机的终止状态: DISCONNECTED

测试步骤：

1. CONNECT\_TEST

2. BEFORE **Theartbeat** RECEIVE Heartbeat XON

3. BEFORE **Theartbeat** RECEIVE Heartbeat XON

4. DISCONNECT\_TEST

5. CONNECT\_TEST

6. DISCONNECT\_TEST

1. 发送挂起队列测试

所需测试：无。

BACnet参考条款: 10.4.10.1和10.4.10.2

目的:验证SendRequest转换是否从TRANSMIT PENDING状态跳出。

测试思想：此测试用例验证以下状态机、状态和转换:

点对点传输状态机

TRANSMIT READY： SendRequest（queuing of a pending transmission）

连接状态机的初始状态: DISCONNECTED

连接状态机的终止状态: DISCONNECTED

测试步骤：

1. CONNECT\_TEST

2. TRANSMIT Data 0,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

3. RECEIVE Data Ack 0 XON

4. BEFORE **Tout** RECEIVE Data 0 | Data 1,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

5. TRANSMIT Data 1,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Vendor\_Identifier

6. RECEIVE Data Ack 1 XON

7. IF (a Data 0 frame was received in step 4) THEN

TRANSMIT Data Ack 0 XON

ELSE

TRANSMIT Data Ack 1 XON

8. BEFORE **Tout** RECEIVE Data 0 | Data 1,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Vendor\_Identifier,

'Property Value' = (the vendor identifier of the IUT)

9. IF (a Data 0 frame was received in step 8) THEN

TRANSMIT Data Ack 0 XON

ELSE

TRANSMIT Data Ack 1 XON

10. DISCONNECT\_TEST

1. 发送挂起未连接测试

所需测试：无。

BACnet参考条款：10.4.10.4。

目的：验证Disconnect transition是否从TRANSMIT PENDING状态退出。

测试思想：这个测试用例验证了以下状态机、状态和转换：

点对点传输状态机

TRANSMIT PENDING：Disconnected

连接状态机的初始状态： DISCONNECTED

连接状态机的终止状态： DISCONNECTED

测试步骤：

1.CONNECT\_TEST

2.TRANSMIT Data 0，

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

3.RECEIVE Data Ack 0 XON

4.BEFORE **Tout** RECEIVE Data 0,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

5.DISCONNECT\_TEST

6.WAIT (**Tresponse** + 2 seconds)

7.CHECK（verify that the BACnet-ComplexACK in step 4 was not retransmitted）

1. 正常传输序列测试

所需测试：无。

BACnet参考条款：10.4.10.1，10.4.10.2和10.4.10.3。

目的：验证IUT是否使用正确的序列发送帧。TD可能需要发送包含需要应答的网络层协议数据单元(NPDU)帧，以实现来自IUT的数据传输。

测试思想：这个测试用例验证了以下状态机、状态和转换：

点对点传输状态机

TRANSMIT IDLE: ConnectionEstablishedXON（sending the Heartbeat XON frame; setting

TxSequenceNumber to zero）

TRANSMIT BLOCKED: PeerReceiverReady

TRANSMIT READY: TransmitMessage（sending appropriately sequenced Data frames）

连接状态机的初始状态： DISCONNECTED

连接状态机的终止状态： DISCONNECTED

测试步骤：

1.CONNECT\_TEST

2.TRANSMIT Data 0,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier'= Object\_Type

3.RECEIVE Data Ack 0 XON

4.BEFORE **Tout** RECEIVE Data 0,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

5.TRANSMIT Data Ack 0 XON

6.TRANSMIT Data 1,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

7.RECEIVE Data Ack 1 XON

8.BEFORE **Tout** RECEIVE Data 1,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

9.TRANSMIT Data Ack 1 XON

10.TRANSMIT Data 0,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

11.RECEIVE Data Ack 0 XON

12.BEFORE **Tout** RECEIVE Data 0,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

13.TRANSMIT Data Ack 0 XON

14.DISCONNECT\_TEST

1. 重传测试

所需测试：无。

BACnet参考条款：10.4.10.3和10.4.10.4。

目的：验证IUT是否正确地重传数据帧。

测试思想：这个测试用例验证了以下状态机、状态和转换：

点对点传输状态机

TRANSMIT READY: TransmitMessage（sending the data frame; RetryCount and ResponseTimer settings）

TRANSMIT PENDING: Retry（resending the data frame; RetryCount and ResponseTimer settings）

TRANSMIT PENDING:RetriesFailed

连接状态机的初始状态： CONNECTED

连接状态机的终止状态： CONNECTED

测试步骤：

1. CONNECT\_TEST

2. TRANSMIT Data 0,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

3. RECEIVE Data Ack 0 XON

4. BEFORE **Tout** RECEIVE Data 0,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

5. WAIT **Tresponse**

6. RECEIVE Data 0,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

7. WAIT **Tresponse**

8. RECEIVE Data 0,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

9. WAIT **Tresponse**

10. RECEIVE Data 0,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

11. WAIT (**Tresponse** + 2 seconds)

12. CHECK (verify that no additional retries were transmitted)

13. TRANSMIT Data 1,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

14. RECEIVE Data Ack 1 XON

15. BEFORE **Tout** RECEIVE Data 1,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

16. WAIT **Tresponse**

17. RECEIVE Data 1,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

18. TRANSMIT Data Ack 1 XON

19. WAIT (**Tresponse** + 2 seconds)

20. CHECK (verify that no additional retries were transmitted)

1. 流控制
2. Heartbeat流控制测试

所需测试：无。

BACnet参考条款：10.4.11.2。

目的：验证IUT正确地应答由TD发送的Heartbeat帧所采用的流控制。

测试思想：TD发送Heartbeat XOFF帧，然后发送一个确认服务请求。TD验证没有发送确认。然后TD发送Heartbeat XON帧并验证应答已发送。这个测试用例验证了以下状态机、状态和转换：

点对点传输状态机

TRANSMIT READY: RemoteBusy

TRANSMIT BLOCKED: SendMessage（message queued for later sending）

点对点接收状态机：

RECEIVE READY: Heartbeat XON（TransmissionBlocked setting）

Heartbeat XOFF（TransmissionBlocked setting）

连接状态机的初始状态： DISCONNECTED

连接状态机的终止状态： DISCONNECTED

测试步骤：

1. CONNECT\_TEST

2. TRANSMIT Heartbeat XOFF

3. TRANSMIT Data 0,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

4. WAIT **Tout**

5. CHECK (verify that the TD does not transmit any messages other than heartbeats)

6. TRANSMIT Heartbeat XON

7. RECEIVE Data Ack 0 XON

8. BEFORE **Tout**RECEIVE Data 0,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

9. DISCONNECT\_TEST

1. 数据确认XOFF流控制测试

所需测试：无。

BACnet参考条款：10.4.11.4。

目的：验证IUT正确地应答由TD发送的Data Ack XOFF帧采用的流控制。

测试思想：TD发送一个确认的服务请求。当接收到应答后，TD发送一个Data Ack XOFF。发送另一个确认传输服务请求以验证没有应答被发送。然后TD发送一个Heartbeat XON并验证应答被发送了。重复该过程以验证接收状态机的两个序列号。这个测试用例验证了以下状态机、状态和转换：

点对点接收状态机

DATA ACK: Ack0\_XOFF（TransmissionBlocked setting）

Ack1\_XOFF（TransmissionBlocked setting）

连接状态机的初始状态： DISCONNECTED

连接状态机的终止状态： DISCONNECTED

测试步骤：

1. CONNECT\_TEST

2. TRANSMIT Data 0,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

3. RECEIVE Data Ack 0 XON

4. BEFORE **Tout** RECEIVE Data 0,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

5.TRANSMIT Data Ack 0 XOFF

6.TRANSMIT Data 1,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

7. WAIT **Tout**

8.CHECK (verify that the TD does not transmit any messages other than heartbeats)

9. TRANSMIT Heartbeat XON

10. RECEIVE Data Ack 1 XON

11. BEFORE **Tout** RECEIVE Data 1,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

12. TRANSMIT Data Ack 1 XOFF

13. TRANSMIT Data 0,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

14. WAIT **Tout**

15. CHECK (verify that the TD does not transmit any messages other than heartbeats)

16. TRANSMIT Heartbeat XON

17. RECEIVE Data Ack 0 XON

18. BEFORE **Tout** RECEIVE Data 0,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

19. DISCONNECT\_TEST

1. 数据Nak XOFF流控制测试

所需测试：无。

BACnet参考条款：10.4.11.5。

目的：验证IUT正确地应答由TD发送的Data Nak XOFF帧采用的流控制。

测试思想：TD发送一个确认服务请求。当接收到应答时，TD发送一个Data Nak XOFF。TD验证IUT没有重试应答。然后TD发送一个Heartbeat XON并验证排队应答重试已发送。重复该过程以验证接收状态机的两个序列号。这个测试用例验证了以下状态机、状态和转换：

点对点接收状态机

DATA ACK: Nak0\_XOFF (TransmissionBlocked setting)

Nak1\_XOFF (TransmissionBlocked setting)

连接状态机的初始状态： DISCONNECTED

连接状态机的终止状态： DISCONNECTED

测试步骤：

1. CONNECT\_TEST

2. TRANSMIT Data 0,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

3.RECEIVE Data Ack 0 XON

4.BEFORE **Tout** RECEIVE Data 0,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

5. TRANSMIT Data Nak 0 XOFF

6. WAIT **Tout**

7. CHECK (verify that the TD does not transmit any messages other than heartbeats)

8. TRANSMIT Heartbeat XON

9. RECEIVE Data 0,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

10. TRANSMIT Data Ack 0 XON

11. TRANSMIT Data 1,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

12. RECEIVE Data Ack 1 XON

13. BEFORE **Tout** RECEIVE Data 1,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

14. TRANSMIT Data Nak 1 XOFF

15. WAIT **Tout**

16. CHECK (verify that the TD does not transmit any messages other than heartbeats)

17. TRANSMIT Heartbeat XON

18. RECEIVE Data 1,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

19. TRANSMIT Data Ack 1 XON

20. DISCONNECT\_TEST

1. 数据Nak XON流控制测试

所需测试：无。

BACnet参考条款：10.4.11.4。

目的：验证IUT正确重试应答Data Nak XON帧。

测试思想：TD发送一个确认服务请求。当应答被接收时，TD发送一个Data Nak XON。TD验证IUT重试应答。重复该过程以验证接收状态机的两个序列号。这个测试用例验证了以下状态机、状态和转换：

点对点接收状态机

RECEIVE READY: DataNak

DATA NAK: Nak0\_XON (TransmissionBlocked setting)

Nak1\_XON (TransmissionBlocked setting)

连接状态机的初始状态： DISCONNECTED

连接状态机的终止状态： DISCONNECTED

测试步骤：

1. CONNECT\_TEST

2. TRANSMIT Data 0,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

3. RECEIVE Data Ack 0 XON

4. BEFORE **Tout** RECEIVE Data 0,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

5. TRANSMIT Data Nak 0 XON

6. RECEIVE Data 0,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

7. TRANSMIT Data 1,

ReadProperty-Request,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type

8. RECEIVE Data Ack 1 XON

9. BEFORE **Tout** RECEIVE Data 1,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

10. TRANSMIT Data Nak 1 XON

11. RECEIVE Data 1,

BACnet-ComplexACK-PDU,

'Object Identifier' = (the IUT's Device object),

'Property Identifier' = Object\_Type,

'Property Value' = DEVICE

12. DISCONNECT\_TEST

1. 接收帧

该部分测试BACnet协议Receive Frame状态机中规定的所有状态和转换。这些测试通常使用Test\_Request和Test\_Response帧来简化测试流程。

特殊状态的出现大多不可测试，状态机包含的正常的路径已经在别的测试中验证了，本部分关注剩下的例外情况。

1. 空闲对空闲测试

所需测试：无。

BACnet参考条款：10.4.7.1。

目的：验证从IDLE状态中选择的转换按照规定正常工作。

测试思想：TD发送一个包含数据的标准的Test\_Request消息并验证来自IUT的应答。然后TD发送一个有错的Test\_Request消息，验证IUT没有应答。接着TD发送一个XOFF(X'13')后跟着发送一个XON(X'11')来验证这些转换。另一个Test\_Request、Test\_Response序列用于验证正确的操作。然后TD发送一些不构成有效帧，不包含X'55'、X'11或X'13'的八位字节。最后，用Test\_Request和Test\_Response序列验证正确操作。

这个测试用例验证了以下状态机、状态和转换：

点对点接收帧状态机

IDLE: EatAnError

EatAnOctet

Flow Control

Preamble1

连接状态机的初始状态： CONNECTED

连接状态机的终止状态： CONNECTED

测试步骤：

1. TRANSMIT Test\_Request,

'Data'= (any valid test data selected by the TD)

2. BEFORE **Tresponse** RECEIVE Test\_Response,

'Data' = (the data transmitted in step 1)

3. TRANSMIT Test\_Request,

'Data' = (any bad data such as the wrong number of data bits, stop bits, or incorrect parity)

4. WAIT **Tresponse**

5. CHECK (verify that no Test\_Response was received)

6. TRANSMIT Test\_Request

7. BEFORE **Tresponse** RECEIVE Test\_Response

8. TRANSMIT X'13'

9. TRANSMIT X'11'

10. TRANSMIT Test\_Request,

'Data' = (any valid test data selected by the TD)

11. BEFORE **Tresponse** RECEIVE Test\_Response,

'Data' = (the data transmitted in step 10)

12. TRANSMIT (any octet string that does not constitute a valid frame and does not contain the octets X”55”, X”11”, or X”13”)

13. TRANSMIT Test\_Request,

'Data' = (any valid test data selected by the TD)

14. BEFORE **Tresponse** RECEIVE Test\_Response,

'Data' = (the data transmitted in step 13)

1. 前导状态对前导状态测试

所需测试：无。

BACnet参考条款：10.4.7.2。

目的：验证IUT正确地从PREAMBLE状态实现所选择的转换。

测试思想：TD发送一个带数据的Test\_Request帧并验证IUT正确地应答。接着，TD发送Preamble1(X'55')，XOFF(X'13')，XON(X'11')和Preamble2(X'FF')，然后发送剩下的包含任意数据的Test\_Request帧。如果IUT对Test\_Request正确地应答，则验证FlowControl、Preamble1和Preamble2转换。在XON之后插入额外的Preamble1重复该过程，则验证了RepeatedPreamble1转换。这个测试用例验证了以下状态机、状态和转换：

点对点接收帧状态机

PREAMBLE: Flow Control

RepeatedPreamble1

Preamble2

连接状态机的初始状态： CONNECTED

连接状态机的终止状态： CONNECTED

测试步骤：

1. TRANSMIT Test\_Request,

'Data' = (any valid test data selected by the TD)

2. BEFORE **Tresponse** RECEIVE Test\_Response,

'Data' = (the data transmitted in step 1)

3. TRANSMIT X”551311FF”,

Test\_Request,

'Data' = (any bad data such as the wrong number of data bits, stop bits, or incorrect parity)

4. BEFORE **Tresponse** RECEIVE Test\_Response,

'Data' = (the data transmitted in step 3)

5. TRANSMIT X”55131155FF”,

Test\_Request,

'Data' = (any bad data such as the wrong number of data bits, stop bits, or incorrect parity)

6. BEFORE **Tresponse** RECEIVE Test\_Response,

'Data' = (the data transmitted in step 5)

1. 前导状态对空闲测试

所需测试：无。

BACnet参考条款：10.4.7.2。

目的：验证IUT正确地实现从PREAMBLE状态所选择的转换。

测试思想：TD发送一个带数据的Test\_Request帧并验证IUT应答正确。接着，TD发送Preamble1，暂停足够长时间使IUT超时，然后继续发送Test\_Request帧。IUT不应该应答，以验证Timeout转换到IDLE状态。然后TD发送一个标准的Test\_Request帧重新验证操作正确。TD发送一个包含错误的Test\_Request帧并验证它是否导致了IDLE状态的转换。TD发送一个标准的Test\_Request帧以重新验证操作正确。TD发送Preamble1，紧接着发送一个非流控制的八位字节或Preamble2，然后是一个标准的Test\_Request帧的剩余部分。最后，TD发送另一个标准的Test\_Request帧以重新验证操作正确。这个测试用例验证了以下状态机、状态和转换：

点对点接收帧状态机

PREAMBLE: Timeout

Error

NotPreamble

连接状态机的初始状态： CONNECTED

连接状态机的终止状态： CONNECTED

测试步骤：

1. TRANSMIT Test\_Request,

'Data' = (any valid test data selected by the TD)

2. BEFORE **Tresponse** RECEIVE Test\_Response,

'Data' = (the data transmitted in step 1)

3. TRANSMIT X'55'

4. WAIT **Tframe\_abort**

5. TRANSMIT X'FF',

Test\_Request,

'Data' = (any valid test data selected by the TD)

6. WAIT **Tresponse**

7. CHECK (verify that no Test\_Response is received)

8. TRANSMIT Test\_Request,

'Data' = (any valid test data selected by the TD)

9. BEFORE **Tresponse** RECEIVE Test\_Response,

'Data' = (the data transmitted in step 8)

10. TRANSMIT X'FF' (any bad data such as the wrong number of data bits, stop bits, or incorrect parity),

Test\_Request,

'Data' = (any valid test data selected by the TD)

11. WAIT **Tresponse**

12. CHECK (verify that no Test\_Response is received)

13. TRANSMIT Test\_Request,

'Data' = (any valid test data selected by the TD)

14. BEFORE **Tresponse** RECEIVE Test\_Response,

'Data' = (the data transmitted in step 13)

15. TRANSMIT X'FF', (any octet except X'FF', X'55', X'13', or X'11'),

Test\_Request,

'Data' = (any valid test data selected by the TD)

16. WAIT **Tresponse**

17. CHECK (verify that no Test\_Response is received)

18. TRANSMIT Test\_Request,

'Data' = (any valid test data selected by the TD)

19. BEFORE **Tresponse** RECEIVE Test\_Response,

'Data' = (the data transmitted in step 18)

1. 头对头测试

所需测试：无。

BACnet参考条款：10.4.7.2。

目的：测试用例验证了一些来自HEADER状态的转换。HEADER状态实际上具有由index变量确定的多个子状态。这些子状态之间的转换无法验证。

测试思想：这个测试用例验证了以下状态机、状态和转换：

点对点接收帧状态机

HEADER: FlowControl

DLE\_Received (DLE\_Mask is handled properly)

HeaderCRC

Length1

Length2

FrameType

连接状态机的初始状态： CONNECTED

连接状态机的终止状态： CONNECTED

测试步骤：

1. TRANSMIT Test\_Request,

'Data' = (any valid test data selected by the TD)

2. BEFORE **Tresponse** RECEIVE Test\_Response,

'Data' = (the data transmitted in step 1)

3. TRANSMIT X”55FF1311”,

Test\_Request,

'Data' = (any valid test data selected by the TD)

4. BEFORE **Tresponse** RECEIVE Test\_Response,

'Data' = (the data transmitted in step 3)

5. TRANSMIT Test\_Request,

'Data' = (any valid test data selected by the TD such that the one of the length octets is X'10', X'11', or X'13')

6. BEFORE **Tresponse**RECEIVE Test\_Response,

'Data' = (the data transmitted in step 1)

1. 头对空闲测试

所需测试：无。

BACnet参考条款：10.4.7.3。

目的：这个测试用例验证了一些来自HEADER状态的转换。

测试思想：这个测试用例验证了以下状态机、状态和转换：

点对点接收帧状态机

HEADER: Timeout

Error

连接状态机的初始状态： CONNECTED

连接状态机的终止状态： CONNECTED

测试步骤：

1. TRANSMIT Test\_Request,

'Data' = (any valid test data selected by the TD)

2. BEFORE **Tresponse** RECEIVE Test\_Response,

'Data' = (the data transmitted in step 1)

3. TRANSMIT X”55FF”

4. WAIT **Tframe\_abort**

5. TRANSMIT (the remainder of a valid Test\_Request frame containing data)

6. WAIT **Tresponse**

7. CHECK (verify that no Test\_Response was transmitted)

8. TRANSMIT Test\_Request,

'Data' = (any valid test data selected by the TD)

9. BEFORE **Tresponse** RECEIVE Test\_Response,

'Data' = (the data transmitted in step 8)

10. TRANSMIT X'55FF'

11. TRANSMIT (any bad data such as the wrong number of data bits, stop bits, or incorrect parity, followed by the remainder of a valid Test\_Request frame)

12. WAIT **Tresponse**

13. CHECK (verify that no Test\_Response was transmitted)

14. TRANSMIT Test\_Request,

'Data' = (any valid test data selected by the TD)

15. BEFORE **Tresponse**RECEIVE Test\_Response,

'Data' = (the data transmitted in step 14)

1. Header\_CRC测试

所需测试：无。

BACnet参考条款：10.4.7.4。

目的：这个测试用例验证了一些来自HEADER\_CRC状态的转换。

测试思想：这个测试用例验证了以下状态机、状态和转换：

点对点接收帧状态机

HEADER\_CRC: BadCRC

FrameTooLong

NoData

Data

连接状态机的初始状态： CONNECTED

连接状态机的终止状态： CONNECTED

测试步骤：

1. TRANSMIT Test\_Request,

'Data' = (any valid test data selected by the TD)

2. BEFORE **Tresponse** RECEIVE Test\_Response,

'Data' = (the data transmitted in step 1)

3. TRANSMIT (a Test\_Request frame that is valid except for a CRC error)

4. WAIT **Tresponse**

5. CHECK (verify that no Test\_Response was transmitted)

6. TRANSMIT Test\_Request,

'Data' = (any valid test data selected by the TD)

7. BEFORE **Tresponse** RECEIVE Test\_Response,

'Data' = (the data transmitted in step 8)

8. IF (the InputBufferSize is known or can be determined) THEN

TRANSMIT Test\_Request,

'Data' = (any valid test data selected by the TD that is sufficiently long to exceed InputBufferSize)

9. WAIT **Tresponse**

10. CHECK (verify that no Test\_Response was transmitted)

11. TRANSMIT Test\_Request,

'Data' = (no data)

12. BEFORE **Tresponse** RECEIVE Test\_Response

1. 数据对数据测试

所需测试：无。

BACnet参考条款：10.4.7.5。

目的：这个测试用例验证了一些来自DATA状态的转换。DATA状态实际上有多个由索引变量决定的子状态。这些子状态之间的转换是无法验证的。

测试思想：这个测试用例验证了以下状态机、状态和转换：

点对点接收帧状态机

DATA: FlowControl

DLE\_Received

DataOctet

CRC1

CRC2

连接状态机的初始状态： CONNECTED

连接状态机的终止状态： CONNECTED

测试步骤：

1. TRANSMIT Test\_Request,

'Data' = (any valid test data selected by the TD)

2. BEFORE **Tresponse** RECEIVE Test\_Response,

'Data' = (the data transmitted in step 1)

3. TRANSMIT Test\_Request,

'Data' = (any valid test data selected by the TD that is interrupted by the flow control character sequence X'1311')

4. BEFORE **Tresponse** RECEIVE Test\_Response,

'Data' = (the data transmitted in step 3)

5. TRANSMIT Test\_Request,

'Data' = (any valid test data selected by the TD that contains the character X'101113' before bit stuffing)

6. BEFORE **Tresponse** RECEIVE Test\_Response,

'Data' = (the data transmitted in step 5)

1. 数据对空闲测试

所需测试：无。

BACnet参考条款：10.4.7.5。

目的：这个测试用例验证了一些来自DATA状态的转换。

测试思想：这个测试用例验证了以下状态机、状态和转换：

点对点接收帧状态机

HEADER: Timeout

Error

连接状态机的初始状态： CONNECTED

连接状态机的终止状态： CONNECTED

测试步骤：

1. TRANSMIT Test\_Request,

'Data' = (any valid test data selected by the TD)

2. BEFORE **Tresponse** RECEIVE Test\_Response,

'Data' = (the data transmitted in step 1)

3. TRANSMIT Test\_Request,

'Data' = (any valid test data with a pause > Tframe\_abort between two octets)

4. WAIT **Tresponse**

5. CHECK (verify that no Test\_Response is received)

6. TRANSMIT Test\_Request,

'Data' = (any valid test data selected by the TD)

7. BEFORE **Tresponse** RECEIVE Test\_Response,

'Data' = (the data transmitted in step 8)

8. TRANSMIT Test\_Request,

'Data' = (any bad data such as the wrong number of data bits, stop bits, or incorrect parity, followed by the remainder of a valid Test\_Request frame)

9. WAIT **Tresponse**

10. CHECK (verify that no Test\_Response is received)

11. TRANSMIT Test\_Request,

'Data' = (any valid test data selected by the TD)

12. BEFORE **Tresponse** RECEIVE Test\_Response,

'Data' = (the data transmitted in step 14)

1. Data\_CRC测试

所需测试：无。

BACnet参考条款：10.4.7.6。

目的：这个测试用例验证了一些来自DATA\_CRC状态的转换。

测试思想：这个测试用例验证了以下状态机、状态和转换：

点对点接收帧状态机

DATA\_CRC: BadCRC

GoodCRC

连接状态机的初始状态： CONNECTED

连接状态机的终止状态： CONNECTED

测试步骤：

1. TRANSMIT Test\_Request,

'Data' = (any valid test data selected by the TD)

2. BEFORE **Tresponse** RECEIVE Test\_Response,

'Data' = (the data transmitted in step 1)

3. TRANSMIT Test\_Request,

'Data' = (any valid test data with bad CRC)

4. WAIT **Tresponse**

5. CHECK (verify that no Test\_Response is received)

6. TRANSMIT Test\_Request,

'Data' = (any valid test data selected by the TD)

7. BEFORE **Tresponse**RECEIVE Test\_Response,

'Data' = (the data transmitted in step 1)

1. 特殊功能测试
   1. 分段

该分段测试检测IUT能否发起和应答分段数据包。若IUT不支持分段，则该测试可省略。

BACnet参考条款：5.2、5.3和5.4。

1. 总体规则和程序

这部分测试需要构造分段的请求和响应，以及在TRANSMIT和RECEIVE语句中使用这些分段。典型使用的有AtomicReadFile、AtomicWriteFile、ReadPropertyMultiple和WritePropertyMultiple服务。任何其他的服务只要请求和应答可以被分成适当的段数，都是可接受的。

除非在测试中明确地说明，否则IUT不能用Abort或Error APDU应答。

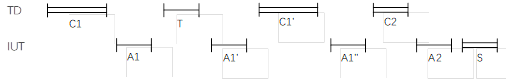
TD将会在测试之间递增Invoke ID。

对于分段数据包，每一个分段的”segmented-message”参数必须设置为TRUE。除了最后一个分段之外，所有分段的”more-follows”参数必须设置为TRUE。初始段的”sequence-number”必须为零，之后每个段依次增加1。参见BACnet应用层条款。

TD将使用”proposed-window-size”为2。IUT被配置为使用”proposed-window-size”为2。

1. 数据包名称

测试使用以下符号来标识序列中的数据包。下图给出了示例。



在这个例子中，TD发起一个请求，而IUT正在应答。在测试期间数据包的序列从左向右进行。

表示从客户端发送的请求数据包的名称以字母C开头。表示从服务器发送的应答数据包的名称以字母S开头。在请求或应答分段的地方，分段号紧跟着前缀。在此图中，C1和C2是完成了分段请求的两个数据包。应答是未分段的且没有数字后缀。

分段确认数据包以字母“A”开头。分段确认数据包后总是跟着一个后缀，该后缀与相应的被确认的分段请求或应答的分段号相匹配。例如，A1是BACnet-SegmentACK-PDU，它的”sequence-number”参数与C1中指定的”sequence-number” 匹配（在本例中，两者都为0）。

当一个数据包被重新发送时，这个数据包名称后跟一个单引号（’）。每次发送相同的数据包时，后面都增加一个单引号。例如，C2，C2’，C2”，代表第一次、第二次和第三次发送的客户端请求第2分段。

当TD通过等待**Segment Fail Time**来模拟被丢弃的数据包时，用字母T来表示，而没有消息被发送。**Segment Fail Time**必须大于或等于TD使用的APDU\_Segment\_Timeout，以给IUT一定的灵活性。

1. TCSL数据包定义

本节测试使用数据包名称作为在TRANSMIT和RECEIVE语句中描述数据包的简称。例如，如果C1和C2是分段请求仅有的两个分段，以下两组TCSL语句是等同的：

TRANSMIT C1

TRANSMIT C2

TRANSMIT BACnet-Confirmed-Request-PDU,

'pdu-type' = 0,

'segmented-message' = TRUE,

'more-follows' = TRUE,

'segmented-response-accepted' = TRUE,

'invokeID' = (available invoke ID),

'sequence-number' = 0,

'proposed-window-size' = 2,

'service-choice' = (service choice used for test),

'service-request' = (first segment of confirmed service request parameters)

TRANSMIT BACnet-Confirmed-Request-PDU,

'pdu-type' = 0,

'segmented-message' = TRUE,

'more-follows' = FALSE,

'segmented-response-accepted' = TRUE,

'invokeID' = (available invoke ID),

'sequence-number' = 1,

'proposed-window-size' = 2,

'service-choice' = (service choice used for test),

'service-request' = (last segment of confirmed service request parameters)

当在TRANSMIT或RECEIVE语句中给定一个PDU参数时，给定的值将覆盖原本为本语句定义的值。例如，以下TCSL语句是等同的：

TRANSMIT C1, 'proposed-window-size' = 1

TRANSMIT BACnet-Confirmed-Request-PDU,

'pdu-type' = 0,

'segmented-message' = TRUE,

'more-follows' = TRUE,

'segmented-response-accepted' = TRUE,

'invokeID' = (available invoke ID),

'sequence-number' = 0,

'proposed-window-size' = 1, -- would otherwise be 2

'service-choice' = (service choice used for test),

'service-request' = (first segment of confirmed service request parameters)

当在RECEIVE语句中BACnet-SegmentACK-PDU被指定时，则假设TD将等待TD的APDU\_Segment\_Timeout属性中设定的值，除非另有指定。例如，以下TCSL语句是等效的：

RECEIVE A3

BEFORE (**Segment Fail Time**)

RECEIVE BACnet-ConfirmedACK-PDU,

'pdu-type' = 4,

'negative-ACK' = FALSE,

'server' = (TRUE when being received from a server),

'original-invokeID' = (invokeID matching request or response),

'sequence-number' = 2, -- this is an ACK of the third segment

'actual-window-size' = (appropriate window size)

1. TD 发起未分段请求和分段应答（Non-Window）

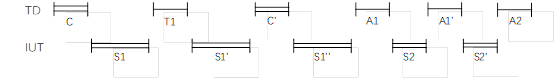
所需测试：无。

目的：测试分段应答的初始传输。

测试思想：等待SegmentTimer和RequestTimer到期是允许IUT假定数据包已丢弃的一种方法。

配置要求：无。

测试步骤：



1. TRANSMIT C

2. RECEIVE S1

3. WAIT **Segment Fail Time**  -- Simulate dropped packet S1

4. RECEIVE S1 -- IUT retransmits (S1’)

5. TRANSMIT C

6. RECEIVE S1 -- IUT retransmits (S1”)

7. TRANSMIT A1

8. RECEIVE S2

9. TRANSMIT A1 -- Simulate dropped packet S2

10. RECEIVE S2 -- IUT retransmits (S2’)

11. TRANSMIT A2

1. TD 发起未分段请求和分段应答（Window）

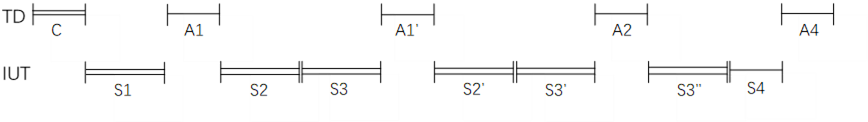
所需测试：无。

目的：测试分段应答中合适的窗管理。

测试思想：强制IUT通过TD响应重传特定的数据包，就好像它们已被丢弃或重新排序一样。

配置要求：IUT的proposed-window-size将被配置为2。

测试步骤：



1. TRANSMIT C

2. RECEIVE S1

3. TRANSMIT A1

4. RECEIVE S2

5. RECEIVE S3

6. TRANSMIT A1 -- Simulate dropped packet S2 or reordered packets S2 and S3 or dropped packets

7. RECEIVE S2 -- IUT retransmits (S2’)

8. RECEIVE S3 -- IUT retransmits (S3’)

9. TRANSMIT A2 -- Simulate dropped packet S3

10. RECEIVE S3 -- IUT retransmits (S3”)

11. RECEIVE S4

12. TRANSMIT A4

如果IUT不能提供被限制在四段的应答数据包，而是扩展到五段或更多，TD可以简单地对剩余段进行应答。

1. TD发起未分段请求和未分段应答(Non-Window)

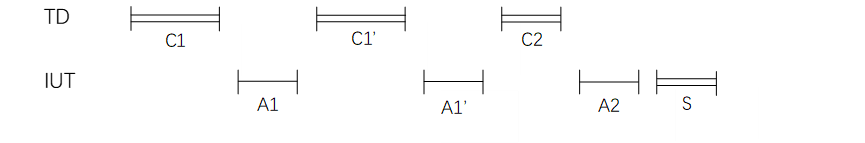
所需测试：无。

目的：验证IUT可以接收到一个简单分段请求。

测试思想：重新发送初始请求数据包。

配置要求：无。

测试步骤：



1. TRANSMIT C1

2. RECEIVE A1

3. TRANSMIT C1 -- Simulate dropped packet A1 (C1’)

4. RECEIVE A1 -- IUT retransmits (A1’)

5. TRANSMIT C2

6. RECEIVE A2

7. RECEIVE S

1. TD发起未分段请求和未分段应答(Window)

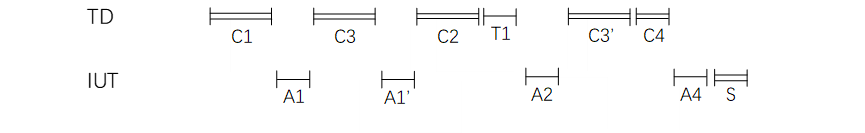
所需测试：无。

目的：验证IUT可以接收一个已被分段的，占用了一个窗口的请求。当该请求被发送时，TD要求一个大小为2的窗口，并且IUT必须以一个大小为2的窗口应答。如果IUT以大小为1的窗口应答，则跳过这个测试。

测试思想：TD发送次序颠倒的请求分段，那么“丢弃”一个分段。

配置要求：无。

测试步骤：



1. TRANSMIT C1

2. RECEIVE A1

3. TRANSMIT C3 -- Simulate dropped packet C2

4. RECEIVE A1 -- Verify C1 was the only packet properly received

5. TRANSMIT C2

6. WAIT **Segment Fail Time** -- Simulate dropped packet C3

7. RECEIVE A2

8. TRANSMIT C3 -- Retransmit packet (C3’)

9. TRANSMIT C4

10. RECEIVE A4

11. RECEIVE S

1. IUT 发起未分段请求和分段应答（Non-Window）

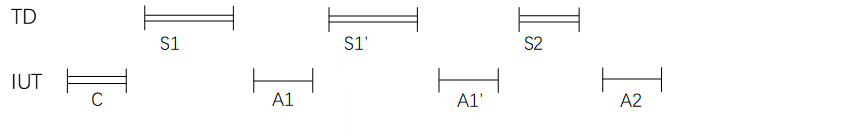
所需测试：无。

目的：验证一个简单分段应答的行为。

测试思想：通过重新发送分段应答的第一个分段，IUT假设第一个ACK被丢弃。

配置要求：无。

测试步骤：



1. MAKE (the IUT initiate the request)

2. RECEIVE C

3. TRANSMIT S1

4. RECEIVE A1

5. TRANSMIT S1 -- Simulate dropped A1 (S1’)

6. RECEIVE A1 -- IUT retransmits (A1’)

7. TRANSMIT S2

8. RECEIVE A2

1. IUT 发起未分段请求和分段应答（Window）

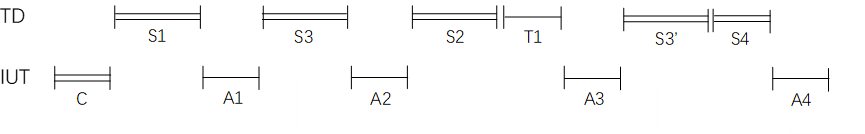
所需测试：无。

目的：验证占用一个窗口的分段应答的行为。TD在初始应答数据包中指定窗口大小为2，IUT返回相同的窗口大小。如果窗口大小为1，那么跳过这个测试。

测试思想：TD发送次序颠倒的应答数据包，那么“丢弃”一个分段。

配置要求：无。

测试步骤：



1. MAKE (the IUT initiate the request)

2. RECEIVE C

3. TRANSMIT S1

4. RECEIVE A1

5. TRANSMIT S3 -- Simulate dropped packet S2

6. RECEIVE A2

7. TRANSMIT S2

8. WAIT **Segment Fail Time** -- Simulate dropped packet S3

9. RECEIVE A3

10. TRANSMIT S3 (S3’)

11. TRANSMIT S4

12. RECEIVE A4

1. IUT发起分段请求和未分段应答（Non-Window）

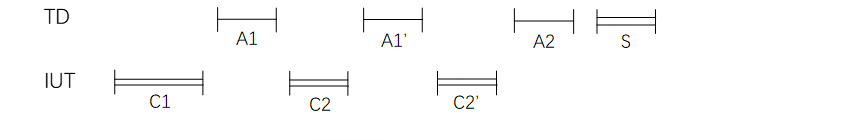
所需测试：无。

目的：验证生成一个分段请求。IUT必须能够发起一个大于TD的Max\_APDU\_Length\_Accepted的最低可配置值的请求。

测试思想：TD重新发送初始的ACK，使IUT假设请求的最后一个分段被丢弃。

配置要求：无。

测试步骤：



1. MAKE (the IUT initiate the request)

2. RECEIVE C1

3. TRANSMIT A1

4. RECEIVE C2

5. TRANSMIT A1 -- Simulate dropped packet C2

6. RECEIVE C2 -- IUT retransmits (C2’)

7. TRANSMIT A2

8. TRANSMIT S1

1. IUT发起分段请求和未分段应答(Window)

所需测试：无。

目的：验证IUT正确地处理生成跨窗口请求。TD在ACK包初始段中指定一个值2。如果请求的初始段指定窗口大小为1，则跳过此测试。

测试思想：测试通过只确认第一个完整窗口的第二分段，验证IUT能正确地重新发送第三分段并继续填充该窗口。

配置要求：无。

测试步骤：



1. MAKE (the IUT initiate the request)

2. RECEIVE C1

3. TRANSMIT A1

4. RECEIVE C2

5. RECEIVE C3

6. TRANSMIT A2 -- Simulate dropped packet C3

7. RECEIVE C3 -- IUT retransmits (C3’)

8. RECEIVE C4

9. TRANSMIT A4

10. TRANSMIT S

1. IUT重新尝试发起分段请求

所需测试：无。

目的：验证在未确认分段请求时IUT能正确处理重试。

测试思想：IUT发起一个分段请求。TD确认第一个分段，然后忽略所有其他消息。IUT重试第一个丢弃的分段，直到此段重试次数超过Number\_Of\_APDU\_Retries。然后IUT将从第一个分段重新开始，并从应答的开始一直重试，直到重试次数超过Number\_Of\_APDU\_Retries。

配置要求:无

测试步骤:

1. MAKE (the IUT initiate the request)

2. RECEIVE C1

3. TRANSMIT A1

4. RECEIVE C2

5. WHILE (segment retries ≤ Number\_Of\_APDU\_Retries) DO {

RECEIVE C2

}

6. WHILE (APDU retries ≤ Number\_Of\_APDU\_Retries) DO {

RECEIVE C1

}

1. 只在超过Max\_APDU\_Length\_Accepted时才分段应答

所需测试：无。

目的：验证IUT应答请求时不分段应答，直到应答的大小超过接收设备接受的Max\_APDU\_Length\_Accepted为止。

测试思想：发送一系列请求，这些请求导致来自IUT的应答连续变长，直到应答超过Max\_APDU\_Length\_Accepted为止。

配置要求：TD应配置为有50个八位字节的Max\_APDU\_Length\_Accepted。

测试步骤：

必须定制测试步骤的详细信息，以匹配IUT的配置。TD须发送一系列ReadProperty或ReadPropertyMultiple请求，这需要越来越长的应答。有些应答应该小于TD的Max\_APDU\_Length\_Accepted，其他的应该大于这个值。

通过结果：

如果应答不超过TD的Max\_APDU\_Length\_Accepted，则IUT应不分段地应答该请求。如果应答超过Max\_APDU\_Length\_Accepted，则IUT应使用分段应答进行应答，每个应答都要尽可能长。

1. 不能分段时终止IUT

本条款定义了以确保IUT在无法使用分段发送长应答时返回正确的终止消息的测试。在BACnet条款5.4.5.3中描述的AWAIT\_RESPONSE Server TSM状态的CannotSendSegmentedComplexACK转换中描述了正确的性能。

如果不支持ReadPropertyMultiple，则可以使用另一个服务生成同样长的应答

1. IUT不支持分段应答

所需测试：无。

目的：验证IUT在不支持分段应答且应答长大于1段时，返回正确的终止消息。

BACnet参考条款：5.4.5.3。

测试思想：TD使用ReadPropertyMultiple请求比单个分段所能容纳的更多的数据。TD还指定应答使用最小的(50个八位字节)分段大小。所请求的数据是IUT的设备对象的Object\_Identifier属性。所请求的数据副本的数量比50字节分段中所能容纳的最大数量多一个。

配置要求:IUT支持执行ReadPropertyMultiple服务，但不支持分段应答的传输。

测试步骤：

1.TRANSMIT ReadPropertyMultiple-Request，

'max-APDU-length-accepted' = B'0000',

'segmented-response-accepted' = TRUE,

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier,

'Property Identifier' = Object\_Identifier,

'Property Identifier' = Object\_Identifier,

'Property Identifier' = Object\_Identifier,

'Property Identifier' = Object\_Identifier

2.RECEIVE BACnet-Abort-PDU，

'Abort Reason' = SEGMENTATION\_NOT\_SUPPORTED

1. TD不支持分段应答

所需测试：无。

目的:验证当请求者不支持分段应答且应答大于一个分段时，IUT返回正确的终止消息。

BACnet参考条款:5.4.5.3。

测试思想:TD使用ReadPropertyMultiple请求比单个分段所能容纳的更多的数据。TD还指定应答使用最小的(50个八位字节)分段大小。所请求的数据是IUT的设备对象的Object\_Identifier属性。所请求的数据副本的数量比50个八位字节分段所能容纳的最大数量多一个。

配置要求:IUT支持执行ReadPropertyMultiple服务，支持分段应答的传输。

测试步骤：

1.TRANSMIT ReadPropertyMultiple-Request，

'max-APDU-length-accepted' = B'0000'

'segmented-response-accepted' = FALSE

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier,

'Property Identifier' = Object\_Identifier,

'Property Identifier' = Object\_Identifier,

'Property Identifier' = Object\_Identifier,

'Property Identifier' = Object\_Identifier

2.RECEIVE BACnet-Abort-PDU

'Abort Reason' = SEGMENTATION\_NOT\_SUPPORTED

1. TD的Max-Segments-Accepted溢出

所需测试：无。

目的：验证IUT在支持分段应答且应答大于TD请求的分段数时返回正确的终止消息。

BACnet参考条款:5.4.5.3 CannotSendSegmentedComplexACK

测试思想：此测试分为两部分。首先，TD请求三段数据，告诉IUT它可以接受四段数据，这应该会成功地收到IUT的一个应答。然后TD请求同样的三段数据，但告诉IUT只接受两段数据，从而导致IUT终止事务。TD使用ReadPropertyMultiple请求数据来填充所需的段数。所请求的数据是IUT的设备对象的Object\_Identifier属性。所请求的数据副本的数量比符合指定段数的最大数量多一个。

配置要求：IUT支持执行ReadPropertyMultiple服务或其他合适的服务，支持分段应答的传输，能够发送50个八位字节大小的两个以上的分段。只有当IUT的协议版本（Protocol\_Revision）大于或等于2时，才进行此测试。

测试方法：如果ReadPropertyMultiple-Request不能与此IUT一起使用，则可以使用另一个合适的服务以生成正确大小的应答。

测试步骤：

1.TRANSMIT ReadPropertyMultiple-Request，

'segmented-response-accepted' = TRUE,

'max-APDU-length-accepted' = B'0000',

'max-segments-accepted' = B'010', -- 4 segments accepted

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier,

'Property Identifier' = Object\_Identifier,

'Property Identifier' = Object\_Identifier,

'Property Identifier' = Object\_Identifier,

'Property Identifier' = Object\_Identifier,

'Property Identifier' = Object\_Identifier,

'Property Identifier' = Object\_Identifier,

'Property Identifier' = Object\_Identifier,

'Property Identifier' = Object\_Identifier,

'Property Identifier' = Object\_Identifier

2. RECEIVE BACnet-ComplexACK-PDU

3. TRANSMIT ReadPropertyMultiple-Request,

'segmented-response-accepted' = TRUE,

'max-APDU-length-accepted' = B'0000',

'max-segments-accepted' = B'001', -- 2 segments accepted

'Object Identifier' = (Device, X),

'Property Identifier' = Object\_Identifier,

'Property Identifier' = Object\_Identifier,

'Property Identifier' = Object\_Identifier,

'Property Identifier' = Object\_Identifier,

'Property Identifier' = Object\_Identifier,

'Property Identifier' = Object\_Identifier,

'Property Identifier' = Object\_Identifier,

'Property Identifier' = Object\_Identifier,

'Property Identifier' = Object\_Identifier,

'Property Identifier' = Object\_Identifier

4. RECEIVE BACnet-Abort-PDU,

'Abort Reason' = BUFFER\_OVERFLOW

* 1. 时间主机

目的：验证一个IUT可以完成时间主机的功能。要成为时间主机，该设备必须能够计时并发出时间同步（TimeSynchronization）服务请求。

所需测试：无。

BACnet参考条款:12.11。

测试思想：时间主机功能要求设备支持设备对象的可选Time\_Synchronization\_Recipient, Local\_Time, Local\_Date, UTC\_Offset 和 Daylight\_Savings\_Status属性。此测试读取一定存在的设备对象的属性，以符合并验证它们是否已更新。要完全完成这些测试，IUT的Time\_Synchronization\_Recipient属性必须列出至少一个有效的接收者。

测试步骤1：TD将使用多个ReadProperty服务请求，读取IUT的设备对象的Local\_Time、Local\_Date、UTC\_Offset、Daylight\_Savings\_Status和Time\_Synchronization\_Receiver属性。

通过结果1：TD将验证对这五个属性中的每个属性，每个读取请求生成一个适当数据类型的值。

测试步骤2：在TD选择的时间间隔(>**Internal Processing Fail Time**)之后，TD将再次读取Local\_Time和Local\_Date属性。

通过结果2：TD应验证时间和日期的新值=测试步骤1中读取的值+TD在此步骤中选择的时间间隔，精度为+/-5%。

测试步骤3：IUT必须成功完成条款8.27的TimeSynchronization服务启动测试。

通过结果3：IUT应按照条款8.27中的要求执行，并将TimeSynchronization服务请求定向到出现在Time\_Synchronization\_Recipient属性中的每个接收者。

测试步骤4：如果PICS表明Local\_Time、Local\_Date和Time\_Synchronization\_Receiver属性是可写的，TD应该将其选择的有效时间、日期和接收者值写入这些属性，并重复上述测试。

通过结果4：这个步骤中新值写入，上面的每一个测试就在成功地完成。

* 1. 字符集

所需测试：条款10.18中的ReadProperty服务执行测试。

BACnet参考条款：20.2.9。

目的：验证IUT支持EPICS中为字符串类型的属性规定的所有字符集。重复进行测试，直到每个支持的字符集都已测试完毕。如果IUT只支持一个字符集，则省略此测试。

测试思想：配置IUT为使用特定的字符集。读取设备对象的Vendor\_Name属性，验证该字符集编码正确。对IUT支持的每个字符集重复该测试过程。

测试步骤：

1. VERIFY (Device, X), Vendor\_Name = (the vendor name)

* 1. 格式错误PDU

BACnet参考条款:18.9和20.1.8。

BACnet要求不处理大多数类型的格式错误的PDU。确认不采取任何措施不在互操作性测试的范围内。但是，异常的确认服务请求会被BACnet-Reject-PDU明确拒绝，拒绝的理由在BACnet 18.9中列出。这组测试生成异常的确认服务请求，以验证它们是否被正确的拒绝。在可能的情况下，这些测试依赖于每个IUT中设备对象的存在以及大多数IUT对读写属性请求的应答能力。

1. 参数不一致

目的：验证IUT可正确应答包含不一致参数的服务请求。

本测试用例见9.15.2.1。

1. 无效参数数据类型

目的：验证IUT对试图将无效数据类型写入属性值作出正确应答。

本测试用例见9.21.2.3。

1. 无效标签

目的：验证IUT正确应答包含无效数据标签的消息。

测试思想：TD发送一个ReadProperty服务请求，该请求的“Property\_Identifier”参数的标签无效。

测试步骤：

1. TRANSMIT ReadProperty-Request,

'Object\_Identifier' = (any object in the IUT's database),

'Property\_Identifier' = (any valid property for the object, but the tag shall have a value X: 2 < X < 254)

2. RECEIVE BACnet-Reject-PDU,

Reject Reason = INVALID\_TAG |

INCONSISTENT\_PARAMETERS |

INVALID\_PARAMETER\_DATA\_TYPE |

MISSING\_REQUIRED\_PARAMETER |

TOO\_MANY\_ARGUMENTS

1. 缺少所需参数

目的：验证IUT正确应答缺少所需参数的消息。

测试思想：TD发送一个ReadProperty服务请求，该请求不包含“Property Identifier”参数。

测试步骤：

1. TRANSMIT ReadProperty-Request,

'Object\_Identifier' = (any object in the IUT's database)

2. RECEIVE BACnet-Reject-PDU,

Reject Reason = MISSING\_REQUIRED\_PARAMETER |

INVALID\_TAG

1. 参数过多

目的：验证IUT正确地应答了包含过多参数的消息。

测试思想：TD发送一个ReadProperty服务请求，该请求包含一个额外的属性标识符。

测试步骤：

1. TRANSMIT ReadProperty-Request,

'Object Identifier' = (any supported object),

'Property Identifier = (any valid property identifier for the specified object),

'Property Identifier' = (any valid property identifier for the specified object not equal to the one in the previous parameter)

2. RECEIVE BACnet-Reject-PDU,

Reject Reason = TOO\_MANY\_ARGUMENTS |

INVALID\_TAG

* 1. 从属代理测试

这些测试验证了IUT可以执行从属代理的功能。为了成为从代理，设备必须能够发现和确认MS/TP从属设备。

BACnet参考条款:12.11.40、12.11.41、12.11.42、12.11.43和16.10.2。

1. 手动从属绑定测试

目的：此测试验证IUT可以在其设备对象的Manual\_Slave\_Binding属性中找到并确认MS/TP从属设备。该测试还验证了IUT正确区分从属设备和主设备，并周期地对从属设备进行确认。

测试思想：使用两个MS/TP设备的地址配置Manual\_Slave\_Binding属性。把其中一个地址设置为从设备，另一个地址设置为支持Who-Is和I-Am服务的主设备。监视网络以验证IUT确认了这两个设备，然后验证从设备地址添加到Slave\_Address\_Binding属性中，而主设备地址没有添加。然后删除从设备，一旦IUT再次确认该从设备，验证从Slave\_Address\_Binding属性中删除该从属设备。

配置要求：MS/TP网络应包含地址A1的从设备(设备标识符为D1)和地址A2的主设备(设备标识符为D2)。从设备不支持使用通配符实例4194303读取其设备对象。主设备应执行Who-Is服务并启动I-Am服务。配置IUT为执行从代理。

测试步骤：

1. BEFORE **Slave Proxy Confirm Interval**

REPEAT addr=(A1, A2) DO {

RECEIVE DESTINATION=addr, SRC=IUT

ReadProperty-Request,

‘Object Identifier’ = (the correct value for the address being queried),

‘Property Identifier’ = Protocol\_Services\_Supported

RECEIVE DESTINATION=IUT, SRC=addr

BACnet-Complex-ACK,

‘Object-Identifier’ = (the correct value for the address being queried),

‘Property Identifier’ = Protocol\_Services\_Supported,

‘Property Value’ = (any valid value for this property)

}

2. VERIFY Slave\_Address\_Binding = ((A1, D1))

3. Remove the slave device from the MS/TP network

4. BEFORE Slave **Proxy Confirm Interval**

RECEIVE DESTINATION=A1, SRC=IUT

ReadProperty-Request,

‘Object Identifier’ = (DEVICE,the correct value for the address being queried),

‘Property Identifier’ = Protocol\_Services\_Supported

-- note that the slave will not reply to this request as it is no longer connected to the network.

5. WAIT (longer than it takes for the IUT to timeout on this request)

6. VERIFY Slave\_Address\_Binding = ()

通过结果：IUT应从每个MS/TP设备读取Protocol\_Services\_Supported属性，以确定其是否支持Who-Is。在IUT确定设备存在于给定的MAC地址之后，实现者可能已选择读取该属性。在这种情况下，在步骤1中任何属性都应被接受，且TD应期望从IUT读取随后附加到MS/TP分段的两个设备的Protocol\_Services\_Supported属性。IUT还可以在测试期间生成它关心的任何其他业务，包括但不限于从它找到的设备读取属性值。

1. 自动从属发现测试

BACnet参考条款:12.11.40、12.11.42、12.11.43和15.5.2。

目的:此测试验证包含Auto\_Slave\_Discovery属性的IUT能够找到并确认支持使用通配符实例4194303读取的MS/TP从属设备。该测试还验证了IUT正确区分从设备和主设备，并周期地对从设备进行确认。

测试思想：配置Auto\_Slave\_Binding属性以启动从设备自动检测。将从设备和主设备连接到MS/TP网络。监视网络以验证IUT是否搜索并找到连接到MS/TP分段的每个设备。验证IUT将从设备添加到其Slave\_Address\_Binding属性中，并且没有将主设备添加到列表中。然后删除一个从设备，一旦IUT再次确认该从设备，将验证从Slave\_Address\_Binding属性中删除该从设备。

配置要求：MS/TP网络应包含地址A1的从设备(设备标识符为D1)和地址A2的主设备(设备标识符为D2)。从设备应支持使用通配符实例4194303读取其设备对象。主设备应执行Who-Is服务并启动I-Am服务。IUT应配置为对除其自身和广播地址外的所有MS/TP地址执行自动从设备检测。

测试步骤：

1.MAKE(the IUT start its automatic slave detection)

2. BEFORE **Slave Proxy Confirm Interval**

REPEAT addr=(all MS/TP addresses excluding the IUT’s MAC address) DO {

RECEIVE DESTINATION=addr, SRC=IUT

ReadProperty-Request,

‘Object Identifier’ = (DEVICE,4194303),

‘Property Identifier’ = Protocol\_Services\_Supported

RECEIVE DESTINATION=IUT, SRC=addr

BACnet-Complex-ACK,

‘Object-Identifier’ = (the correct value for the address being queried),

‘Property Identifier’ = Protocol\_Services\_Supported,

‘Property Value’ = (any valid value for this property)

}

3. VERIFY Slave\_Address\_Binding = ((A1, D1))

4. Remove the slave device from the MS/TP network

5. BEFORE **Slave Proxy Confirm Interval**

RECEIVE DESTINATION=A1, SRC=IUT

ReadProperty-Request,

‘Object Identifier’ = (DEVICE, the correct value for the address being queried),

‘Property Identifier’ = Protocol\_Services\_Supported

-- note that the slave will not reply to this request as it is no longer connected to the network.

6. WAIT (longer than it takes the IUT to timeout on this request)

7. VERIFY Slave\_Address\_Binding = ()

通过结果：IUT应从每个MS/TP设备读取Protocol\_Services\_Supported属性，以确定其是否支持Who-Is。在IUT确定设备存在于给定的槽中之后，实现者可能已经选择读取此属性。在这种情况下，在步骤1中任何属性都应被接受，TD应期望随后从IUT读取附加到MS/TP分段的两个设备的Protocol\_Services\_Supported属性。IUT还可以在测试期间生成它关心的任何其他业务，包括但不限于从它找到的设备读取属性值。

1. 代理测试

BACnet参考条款:12.11.40、12.11.41、12.11.42、12.11.43和16.10.2。

目的：此测试验证IUT能正确代理Slave\_Address\_Binding属性中列出的从设备。

测试思想：配置IUT，使其代理从设备，等待IUT找到并确认从设备。发出Who-Is所有形式的请求，以确保IUT正确代理从设备的I-Am应答。将TD连接到MS/TP分段，并将TD连接到另一个BACnet网络，重复测试。

配置要求：MS/TP网络应包含地址A1的从设备（设备标识符为D1）。IUT应配置为执行从代理。测试在IUT成功找到并确认从设备后开始。此测试应在连接到MS/TP网络的TD上重复一次，并在连接到另一个BACnet网络的TD上重复一次。

测试步骤：

1. TRANSMIT DESTINATION = GLOBAL BROADCAST, Who-Is

2. BEFORE **Unconfirmed Response Fail Time**

RECEIVE

DESTINATION = GLOBAL BROADCAST | LOCAL BROADCAST | TD

SOURCE = A1

I-Am-Request,

'I Am Device Identifier' = (the slave’s Device object’s Object\_Identifer),

'Max APDU Length Accepted' = (the slave’s value for this property),

'Segmentation Supported' = FALSE,

'Vendor Identifier' = (the slave’s value for this property)

3. TRANSMIT DESTINATION = A1, Who-Is

4. BEFORE **Unconfirmed Response Fail Time**

RECEIVE

DESTINATION = GLOBAL BROADCAST | LOCAL BROADCAST | TD

SOURCE = A1

I-Am-Request,

'I Am Device Identifier' = (the slave’s Device object’s Object\_Identifer),

'Max APDU Length Accepted' = (the slave’s value for this property),

'Segmentation Supported' = FALSE,

'Vendor Identifier' = (the slave’s value for this property)

5. TRANSMIT DESTINATION = GLOBAL BROADCAST, Who-Is

6. BEFORE **Unconfirmed Response Fail Time**

RECEIVE

DESTINATION = GLOBAL BROADCAST | LOCAL BROADCAST | TD

SOURCE = A1

I-Am-Request,

'I Am Device Identifier' = (the slave’s Device object’s Object\_Identifer),

'Max APDU Length Accepted' = (the slave’s value for this property),

'Segmentation Supported' = FALSE,

'Vendor Identifier' = (the slave’s value for this property)

* 1. 自动网络映射

目的：验证IUT可以找到所有连接到BACnet互连网络的设备，并向用户显示设备列表。

测试配置：IUT应连接到由分布在多个网络上的设备组成的网络。TD应监视IUT生成的Who-Is服务请求，并应验证这些请求涵盖BACnet设备实例的完整范围，且所使用的Who-Is请求包含设备实例范围。

测试步骤：

1. IF (the IUT caches device information) THEN

MAKE (the IUT’s cache clear so that is unaware of the existence of any other devices)

2. MAKE (the IUT initiate the network mapping function while monitoring the network for Who-Is requests initiated by the IUT)

3. CHECK (that the IUT identifies all of the devices on the network to the user)

通过结果：IUT发送全球广播的Who-Is请求或向每个网络定向广播，Who-Is请求覆盖整个设备示例范围，IUT不单独依赖于没有设备示例范围的Who-Is请求。

通过结果：IUT向用户显示网络上所有设备的列表。该列表不应指明目前在互连网络上不存在的设备。IUT自身不需要包含在列表中。

* 1. 自动设备映射

目的：验证IUT可以在任意BACnet设备中找到所有对象，并将这些对象呈现给用户。

配置：将TD配置为模拟具有任意一组自一致BACnet特征的设备，其中包含一组对象。

测试步骤：

1. MAKE (the IUT initiate its device mapping function for the TD)

2. CHECK (that the IUT correctly identifies all of the objects in the TD)

1. IP功能测试

此条款定义了证明BACnet/IP功能所需的测试，如BACnet标准附件J中定义的那样。对于每个测试用例，将描述一个或多个将要交换的消息的序列。当IUT和TD完全按照测试用例中描述的那样交换消息时，产生测试结果。任何其他消息组合都导致测试的失败。一些测试用例是无效的，除非本标准中定义的其他测试已经执行，并且IUT通过了这个测试。这些所需测试在测试用例描述中说明。

对于本条款中的测试，DESTINATION是引用设备的B/IP地址。例如，DESTINATION = FD2表示DESTINATION = (FD2的B/IP地址)。

* 1. Non-BBMD B/IP设备

这组测试验证非BACnet广播管理设备(BBMD)的B/IP设备将正确应答与BBMD相关的传入B/IP消息。只有不支持(或配置为不支持)BBMD功能的设备才能执行这些测试。

1. Write-Broadcast-Distribution-Table写入广播分发表

目的:验证未配置为BBMD的IUT将拒绝Write-Broadcast-Distribution-Table请求。

测试步骤：

1. TRANSMIT DESTINATION = IUT, SOURCE = TD,

Write-Broadcast-Distribution-Table

2. RECEIVE DESTINATION = TD, SOURCE = IUT,

BVLC-Result,

'Result Code' = Write-Broadcast-Distribution-Table NAK

1. Read-Broadcast-Distribution-Table读取广播分发表

目的：验证未配置为BBMD的IUT将拒绝Read-Broadcast-Distribution-Table请求。

测试步骤：

1. TRANSMIT DESTINATION = IUT, SOURCE = TD,

Read-Broadcast-Distribution-Table

2. RECEIVE DESTINATION = TD, SOURCE = IUT,

BVLC-Result,

'Result Code' = Read-Broadcast-Distribution-Table NAK

1. 注册外部设备

目的：验证未配置为BBMD的IUT将拒绝Register-Foreign-Device请求。

测试步骤：

1. TRANSMIT DESTINATION = IUT, SOURCE = TD,

Register-Foreign-Device

2. RECEIVE DESTINATION = TD, SOURCE = IUT,

BVLC-Result,

'Result Code' = Register-Foreign-Device NAK

1. Delete-Foreign-Device-Entry删除外部设备入口

目的:验证未配置为BBMD的IUT将拒绝Delete-Foreign-Device的请求。

测试步骤：

1. TRANSMIT DESTINATION = IUT, SOURCE = TD,

Delete-Foreign-Device

2. RECEIVE DESTINATION = TD, SOURCE = IUT,

BVLC-Result,

'Result Code' = Delete-Foreign-Device NAK

1. Read-Foreign-Device-Table读取外部设备表

目的:验证未配置为BBMD的IUT将拒绝Read-Foreign-Device-Table的请求。

测试步骤：

1. TRANSMIT DESTINATION = IUT, SOURCE = TD,

Read-Foreign-Device-Table

2. RECEIVE DESTINATION = TD, SOURCE = IUT,

BVLC-Result,

'Result Code' = Read-Foreign-Device-Table NAK

1. 向网络分发广播Distribute-Broadcast-To-Network

目的:验证未配置为BBMD的IUT将拒绝Distribute-Broadcast-To-Network的请求。

测试步骤：

1. TRANSMIT DESTINATION = IUT, SOURCE = TD,

Distribute-Broadcast-To-Network,

NPDU = Who-Is

2. RECEIVE DESTINATION = TD, SOURCE = IUT,

BVLC-Result,

'Result Code' = Distribute-Broadcast-To-Network NAK

1. 转发NPDU (单跳分发)

目的：验证未配置为BBMD的IUT将处理Forwarded-NPDU消息。

配置要求：IUT不应配置为BBMD。TD应位于与IUT不同的IP子网上。

测试步骤：

1. TRANSMIT DA = Directed IP Broadcast to IUT”s IP Subnet,SOURCE = TD,

Forwarded-NPDU,

Originating-Device = TD,

NPDU = Who-Is

2. RECEIVE DA = Local IP Broadcast, SOURCE = IUT,

Original-Broadcast-NPDU,

NPDU = I-Am

3. CHECK (The IUT shall not issue any Forwarded-NPDUs)

1. 原始广播NPDU

目的：验证未配置为BBMD的IUT将处理Original-Broadcast-NPDU消息。

测试步骤：

1. TRANSMIT DA = Local IP Broadcast, SOURCE = TD,

Original-Broadcast-NPDU,

NPDU = Who-Is

2. RECEIVE DA = Local IP Broadcast, SOURCE = IUT,

Original-Broadcast-NPDU,

NPDU = I-Am

3. CHECK (The IUT shall not issue any Forwarded-NPDUs)

1. 原单播NPDU

目的：验证未配置为BBMD的IUT将处理Original-Unicast-NPDU消息。

测试步骤：

1. TRANSMIT DESTINATION = IUT, SOURCE = TD,

Original-Unicast-NPDU,

NPDU = BACnet ReadProperty-Request

2. RECEIVE DESTINATION = TD, SOURCE = IUT,

Original-Unicast-NPDU,

NPDU = ReadProperty-ACK

1. 转发NPDU (两跳分发)

目的：验证未配置为BBMD的IUT将处理Forwarded-NPDU消息。

配置要求：IUT不应配置为BBMD。TD应与IUT在同一子网上。D1是与TD在不同的IP子网上的设备。

测试步骤：

1. TRANSMIT DA = Local IP Broadcast, SOURCE = TD,

Forwarded-NPDU,

Originating-Device = D1,

NPDU = Who-Is

2. RECEIVE DA = Local IP Broadcast, SOURCE = IUT,

Original-Broadcast-NPDU,

NPDU = I-Am

3. CHECK (The IUT shall not issue any Forwarded-NPDUs)

* 1. 带有服务器应用程序的BBMD B/IP设备

这组测试验证了带有服务器应用程序的BBMD B/IP设备将正确地处理Forwarded-NPDU, Original-Broadcast-NPDU和Original-Unicast-NPDU消息中的NPDU。

配置要求：在IUT中运行服务器应用程序。对于单跳分发测试，图15-1中的互联网路由器必须配置为转发定向广播。

注意事项：图15-1显示了测试15.2-15.7的逻辑网络配置。测试不需要完整的网络，只要IUT能够接收到来自指定设备的数据包，指定TD在每个测试中执行TRANSMIT语句时的角色。TD还必须在所有测试过程中监控IUT的子网，RECEIVE的内容是来自IUT的子网。为了实现这一点，TD可以是多宿主的，或者可以使用另一个TD来监视IUT的子网。

本地IP广播意味着目标IP地址的主机部分都是1，MAC层的目标地址也是一个广播。定向IP广播是指目标IP地址的主机部分都是1，MAC层的目标地址等于路由器的MAC地址。IP地址的主机部分是子网掩码中为0的那些位。测试人员应为每台设备选择合适的IP地址和子网掩码。



**图15-1.** BBMD测试的逻辑网络配置

1. 执行转发NPDU

目的:验证IUT将向其应用程序实体传递Forwarded-NPDU消息。

配置要求:TD扮演BBMD1的角色。步骤1中不包含网络层报头中的DNET/DADR和SNET/SADR字段。

该测试分为单跳分发测试和两跳分发测试。

1. 执行转发NPDU(单跳分发)

配置要求:IUT应配置为BDT, BDT包括:

|  |  |
| --- | --- |
| B/IP地址 | 广播分发掩码 |
| IUT | IP子网1 子网掩码 |
| BBMD1 | IP子网2 子网掩码 |

测试步骤：

1.TRANSMIT

DA = Directed IP Broadcast to IP Subnet 1,

SA = BBMD1,

Forwarded-NPDU,

Originating-Device = BBMD1,

NPDU = Who-Is

2.RECEIVE

DA = Local IP Broadcast,

SA = IUT,

Original-Broadcast-NPDU,

NPDU = I-Am

3.RECEIVE

DA = Directed IP Broadcast to IP Subnet 2,

SA = IUT,

Forwarded-NPDU,

Originating-Device = IUT,

NPDU = I-Am

4. CHECK (The IUT does not forward or resend the Who-Is packet out the port on which it was received)

测试人员注意:IUT发送的消息的顺序并不重要。

1. 执行转发NPDU(两跳分发)

配置要求:IUT应配置为BDT, BDT包括:

|  |  |
| --- | --- |
| B/IP地址 | 广播分发掩码 |
| IUT | 255.255.255.255 |
| BBMD1 | 255.255.255.255 |

测试步骤：

1.TRANSMIT

DA = IUT,

SA = BBMD1,

Forwarded-NPDU,

Originating-Device = BBMD1,

NPDU = Who-Is

2.RECEIVE

DA = Local IP Broadcast,

SA = IUT,

Forwarded-NPDU,

Originating-Device = BBMD1,

NPDU = Who-Is

3.RECEIVE

DA = Local IP Broadcast,

SA = IUT,

Original-Broadcast-NPDU,

NPDU = I-Am

4.RECEIVE

DA = BBMD1,

SA = IUT,

Forwarded-NPDU,

Originating-Device = IUT,

NPDU = I-Am

测试人员注意:：IUT发送的消息的顺序并不重要。

1. 执行Original-Broadcast-NPDU原始广播NPDU

目的：验证IUT将Original-Broadcast-NPDU消息传递给其应用实体。

配置要求：TD为设备D1。步骤1中不包含网络层报头中的DNET/DADR和SNET/SADR字段。

该测试分为单跳分发测试和两跳分发测试。

1. Original-Broadcast-NPDU原始广播NPDU (单跳分发)

配置要求:IUT应配置为BDT, BDT包括:

|  |  |
| --- | --- |
| B/IP地址 | 广播分发掩码 |
| IUT | IP子网1 子网掩码 |
| BBMD1 | IP子网2 子网掩码 |

测试步骤：

1.TRANSMIT

DA = Local IP Broadcast,

SA = D1,

Original-Broadcast-NPDU,

NPDU = Who-Is

2. RECEIVE

DA = Directed IP Broadcast to IP Subnet 2,

SA = IUT,

Forwarded-NPDU,

Originating-Device = D1,

NPDU = Who-Is

3. RECEIVE

DA = Local IP Broadcast,

SA = IUT,

Original-Broadcast-NPDU,

NPDU = I-Am

4. RECEIVE

DA = Directed IP Broadcast to IP Subnet 2,

SA = IUT,

Forwarded-NPDU,

Originating-Device = IUT,

NPDU = I-Am

测试人员注意：IUT发送的消息的顺序并不重要。

1. 执行原始广播NPDU (两跳分发)

配置要求：IUT应配置为BDT, BDT包括：

|  |  |
| --- | --- |
| B/IP地址 | 广播分发掩码 |
| IUT | 255.255.255.255 |
| BBMD1 | 255.255.255.255 |

测试步骤：

1. TRANSMIT

DA = Local IP Broadcast,

SA = D1,

Original-Broadcast-NPDU,

NPDU = Who-Is

2. RECEIVE

DA = BBMD1,

SA = IUT,

Forwarded-NPDU,

Originating-Device = D1,

NPDU = Who-Is

3. RECEIVE

DA = Local IP Broadcast,

SA = IUT,

Original-Broadcast-NPDU,

NPDU = I-Am

4. RECEIVE

DA=BBMD1,

SA=IUT,

Forwarded-NPDU,

Originating-Device = IUT,

NPDU = I-Am

测试人员注意：IUT发送的消息的顺序并不重要。

1. 执行Original-Unicast-NPDU原单播NPDU

目的：验证IUT将向其应用程序实体传递Original-Unicast-NPDU消息。

配置要求：TD为设备D1。步骤1中不包含网络层报头中的DNET/DADR和SNET/SADR字段。IUT应配置为BDT，该BDT包含：

|  |  |
| --- | --- |
| B/IP地址 | 广播分发掩码 |
| IUT | 255.255.255.255 |
| BBMD1 | 255.255.255.255 |

测试步骤：

1. TRANSMIT

DA = IUT,

SA = D1,

Original-Unicast-NPDU,

NPDU = Read-Property

2. RECEIVE

DA = D1,

SA = IUT,

Original-Unicast-NPDU,

NPDU = Read-Property-Ack

3. CHECK(The IUT does not forward to BBMD1 either packet from step 1 or step 2)

* 1. 广播分发表操作

这组测试验证没有FDT的BACnet广播管理设备将正确执行BDT操作。

配置要求:TD为设备D1。IUT应配置为BDT，该BDT包含:

|  |  |
| --- | --- |
| B/IP地址 | 广播分发掩码 |
| IUT | 255.255.255.255 |
| BBMD1 | 255.255.255.255 |

IUT应通过其启动程序。

1. 执行Write-Broadcast-Distribution-Table写广播分发表(表增长)

目的:验证配置为BBMD的IUT在新表大于当前表时将执行Write-Broadcast-Distribution-Table 请求。

配置要求:IUT按照14.3的要求配置。

测试步骤：

1. TRANSMIT

DA = IUT,

SA = D1,

Write-Broadcast-Distribution-Table,

(List of BDT Entries consisting of

BBMD1 255.255.255.255

BBMD2 255.255.255.255

BBMD3 255.255.255.255

IUT 255.255.255.255

)

2. RECEIVE

DA = D1,

SA = IUT,

BVLC-Result message,

'Result Code' = Successful completion

3. TRANSMIT

DA = IUT,

SA = D1,

Read-Broadcast-Distribution-Table

4. RECEIVE

DA = D1,

SA = IUT,

Read-Broadcast-Distribution-Table-Ack,

List of BDT Entries

5. CHECK(List of BDT Entries consisting of four entries (order unspecified)

BBMD1 255.255.255.255

BBMD2 255.255.255.255

BBMD3 255.255.255.255

IUT 255.255.255.255

）

1. 执行Write-Broadcast-Distribution-Table写广播分发表(表增长)

目的：验证配置为BBMD的IUT在新表小于当前表时将执行Write-Broadcast-Distribution-Table请求。

配置要求：IUT的BDT至少有4个条目。

测试步骤：

1. TRANSMIT

DA = IUT,

SA = D1,

Write-Broadcast-Distribution-Table,

(List of BDT entries consisting of three entries

BBMD2 255.255.255.255

BBMD3 255.255.255.255

IUT 255.255.255.255

)

2. RECEIVE

DA = D1,

SA = IUT,

BVLC-Result,

'Result Code' = Successful completion

3. TRANSMIT

DA = IUT,

SA = D1,

Read-Broadcast-Distribution-Table

4. RECEIVE

DA = D1,

SA = IUT,

Read-Broadcast-Distribution-Table-Ack,

List of BDT entries

5. CHECK(List of BDT entries consisting of three entries (order unspecified)

BBMD2 255.255.255.255

BBMD3 255.255.255.255

IUT 255.255.255.255

)

1. 验证从上一会话中保存的配置创建的广播分发表

目的：验证BBMD将更新本地配置数据库中的BDT，并在启动时对其进行初始化。

配置要求：IUT的BDT不包含步骤1中写的同样的条目。

测试步骤：

1.TRANSMIT

DA = IUT,

SA = D1,

Write-Broadcast-Distribution-Table,

(List of BDT entries consisting of three entries

IUT 255.255.255.255

BBMD1 255.255.255.255

BBMD2 255.255.255.255

)

2. RECEIVE

DA = D1,

SA = IUT,

BVLC-Result,

'Result Code' = Successful completion

3. WAIT ( Vendor specified period for BDT to be saved in non-volatile memory)

4. MAKE (the IUT reset)

5. TRANSMIT

DA = IUT,

SA = D1,

Read-Broadcast-Distribution-Table

6. RECEIVE

DA = D1,

SA = IUT,

Read-Broadcast-Distribution-Table-Ack,

List of BDT Entries

7. CHECK(List of BDT Entries consisting of three entries (order unspecified)

IUT 255.255.255.255

BBMD1 255.255.255.255

BBMD2 255.255.255.255

)

1. Write-Broadcast-Distribution-Table写广播分发表(空表)

目的：验证配置为BBMD的IUT将拒绝包含空表的Write-Broadcast-Distribution-Table请求，因为它不包含可用于IUT的条目。

测试步骤：

1. TRANSMIT DESTINATION = IUT, SOURCE = TD,

Write-Broadcast-Distribution-Table,

Empty BDT

2. RECEIVE DESTINATION = TD, SOURCE = IUT,

BVLC-Result,

'Result Code' = Write-Broadcast-Distribution-Table NAK

3. TRANSMIT DESTINATION = IUT, SOURCE = TD,

Read-Broadcast-Distribution-Table

4. RECEIVE DESTINATION = TD, SOURCE = IUT,

Read-Broadcast-Distribution-Table-Ack,

List of BDT Entries

5. CHECK(List of BDT Entries consisting of three entries (order unspecified)

(123.4.5.6: 0xBAC0) 255.0.0.0

(123.7.8.9: 0xBAC4) 255.0.0.0

(123.7.8.9: 0xBAC5) 255.255.0.0

)

1. Write-Broadcast-Distribution-Table写广播分发表(不包含BBMD条目)

目的：验证配置为BBMD的IUT将拒绝Write-Broadcast-Distribution-Table请求，如果正在写入的表中不包含此BBMD条目。

测试步骤：

1. TRANSMIT DESTINATION = IUT, SOURCE = TD,

Write-Broadcast-Distribution-Table,

BDT (consisting of two entries

(123.7.8.9: 0xBAC6) 255.255.0.0

(123.7.8.9: 0xBAC7) 255.255.255.0

)

2. RECEIVE DESTINATION = TD, SOURCE = IUT,

BVLC-Result,

'Result Code' = Write-Broadcast-Distribution-Table NAK

3. TRANSMIT DESTINATION = IUT, SOURCE = TD,

Read-Broadcast-Distribution-Table

4. RECEIVE DESTINATION = TD, SOURCE = IUT,

Read-Broadcast-Distribution-Table-Ack,

List of BDT Entries

5. CHECK(List of BDT Entries consisting of three entries (order unspecified)

(123.4.5.6: 0xBAC0) 255.0.0.0

(123.7.8.9: 0xBAC4) 255.0.0.0

(123.7.8.9: 0xBAC5) 255.255.0.0

)

1. 验证从上一会话中保存的配置创建的广播分发表

目的:验证没有FDT的BBMD将更新本地配置数据库中的BDT，并在启动时对其进行初始化。

配置要求:IUT应按照完成测试14.3.5的结果进行配置(或处于该状态)。

测试步骤：

1. TRANSMIT DESTINATION = IUT, SOURCE = TD,

Read-Broadcast-Distribution-Table

2. RECEIVE DESTINATION = TD, SOURCE = IUT,

Read-Broadcast-Distribution-Table-Ack,

List of BDT Entries

3. CHECK(List of BDT Entries consisting of three entries (order unspecified)

(123.4.5.6: 0xBAC0) 255.0.0.0

(123.7.8.9: 0xBAC4) 255.0.0.0

(123.7.8.9: 0xBAC5) 255.255.0.0

)

* 1. 外部设备表操作（负响应测试）

这组测试验证没有外部设备表的BACnet广播管理设备将拒绝传入外部设备表请求。对于本节中的所有测试，从IUT的角度来看，TD是一个外部设备。

1. 注册外部设备

目的：配置IUT为没有FDT的BBMD，并验证IUT将拒绝Register-Foreign-Device请求。

测试步骤：

1. TRANSMIT DESTINATION = IUT, SOURCE = TD,

Register-Foreign-Device,

'Time-to-Live' = any integer value greater than 0 and less than 65,536

2. RECEIVE DESTINATION = TD, SOURCE = IUT,

BVLC-Result,

'Result Code'= Register-Foreign-Device NAK

1. Delete-Foreign-Device删除外部设备

目的：配置IUT为没有FDT的BBMD，验证IUT将拒绝Delete-Foreign-Device的请求。

测试步骤:

1. TRANSMIT DESTINATION = IUT, SOURCE = TD,

Delete-Foreign-Device-Table-Entry,

FDT Entry' = any valid B/IP address

2. RECEIVE DESTINATION = TD, SOURCE = IUT,

BVLC-Result,

'Result Code' = Delete -Foreign-Device-Table-Entry N

1. Read-Foreign-Device-Table

目的：验证IUT配置为没有FDT的BBMD，将拒绝Read-Foreign-Device-Table请求。

测试步骤：

1.TRANSMIT DESTINATION = IUT, SOURCE = TD,

Read-Foreign-Device-Table

2. RECEIVE DESTINATION = TD, SOURCE = IUT,

BVLC-Result,

'Result Code' = Read-Foreign-Device-Table NAK

1. Distribute-Broadcast-To-Network

目的：验证IUT配置为没有FDT的BBMD，将拒绝Distribute-Broadcast-To-Network请求。

测试步骤：

1.TRANSMIT DESTINATION = IUT, SOURCE = TD,

Distribute-Broadcast-To-Network,

'BACnet NPDU from Originating Device' = any well-formed NPDU

2. RECEIVE DESTINATION = TD, SOURCE = IUT,

BVLC-Result,

'Result Code' = Distribute-Broadcast-To-Network NAK

* 1. BACnet广播管理（无外部设备表，无应用程序）

这组测试验证了一个没有外部设备表的BACnet广播管理设备将正确处理Forwarded-NPDU, Original-Broadcast-NPDU和Original-Unicast-NPDU消息。

1. Forwarded-NPDU消息被忽略

目的：验证配置IUT为没有FDT的BBMD时，可处理Forwarded-NPDU消息。

测试步骤：

1.TRANSMIT DESTINATION = IUT, SOURCE = TD,

Forwarded-NPDU,

NPDU = Who-Is

2. CHECK (The IUT shall not take any action)

1. Original-Broadcast-NPDU消息转发

目的：验证IUT配置为没有FDT的BBMD时，可将处理一个Original-Broadcast-NPDU消息。

配置要求：TD在IUT的子网上应是设备D1。在步骤1中不应该出现网络层报头中的DNET/DADR和SNET/SADR字段。

该测试被分为单跳分发和两跳分发两个测试。

1. Original-Broadcast-NPDU消息转发（单跳分发）

配置要求：IUT应配置一个BDT，BDT包含：

|  |  |
| --- | --- |
| B/IP 地址 | 广播分发掩码 |
| IUT | IP子网1 子网掩码 |
| BBMD1 | IP子网2 子网掩码 |
| BBMD2 | IP子网3 子网掩码 |
| BBMD3 | IP子网4 子网掩码 |

测试步骤：

1. TRANSMIT

DA = Local IP Broadcast,

SA = D1,

Original-Broadcast-NPDU,

NPDU = Who-Is

2. RECEIVE

DA = Directed IP Broadcast to IP Subnet 2,

SA = IUT,

Forwarded-NPDU,

Originating-Device = D1,

NPDU = Who-Is

3. RECEIVE

DA = Directed IP Broadcast to IP Subnet 3,

SA = IUT,

Forwarded-NPDU,

Originating-Device = D1,

NPDU = Who-Is

4. RECEIVE

DA = Directed IP Broadcast to IP Subnet 4,

SA = IUT,

Forwarded-NPDU,

Originating-Device = D1,

NPDU = Who-Is

注意事项：IUT发送的消息顺序并不重要。

1. Original-Broadcast-NPDU消息转发（两跳分发）

配置要求：IUT应配置一个BDT，其中BDT包含：

|  |  |
| --- | --- |
| B/IP 地址 | 广播分发掩码 |
| IUT | 255.255.255.255 |
| BBMD1 | 255.255.255.255 |
| BBMD2 | 255.255.255.255 |
| BBMD3 | 255.255.255.255 |

测试步骤：

1. TRANSMIT

DA = Local IP Broadcast,

SA = D1,

Original-Broadcast-NPDU,

NPDU = Who-Is

2. RECEIVE

DA = BBMD1,

SA = IUT,

Forwarded-NPDU,

Originating-Device = D1,

NPDU = Who-Is

3. RECEIVE

DA = BBMD2,

SA = IUT,

Forwarded-NPDU,

Originating-Device = D1,

NPDU = Who-Is

4. RECEIVE

DA = BBMD3,

SA = IUT,

Forwarded-NPDU,

Originating-Device = D1,

NPDU = Who-Is

注意事项：IUT发送的消息顺序并不重要。

1. Original-Unicast-NPDU消息被忽略

目的：验证IUT配置为没有FDT的BBMD时，将处理一个Original-Unicast-NPDU消息。

测试步骤：

1. TRANSMIT DESTINATION = IUT, SOURCE = TD,

Original-Unicast-NPDU,

NPDU = Read-Property

2. CHECK (The IUT shall not take any action)

* 1. 外部设备管理

这组测试验证具有FDT的BBMD将正确执行FDT操作。设FD1为外部设备，其B/IP地址为123.5.6.7：0xBAC0；FD2为外部设备，其B/IP地址为123.5.6.8：0xBAC0。

配置要求：在执行此组测试之前，应配置IUT使BBMD和FDT被选中。IUT应通过其启动程序，不得运行任何应用程序。

1. 执行Read-Foreign-Device-Table

目的：验证IUT配置有一个空FDT，将处理一个Read-Foreign-Device-Table请求。

测试步骤：

1. TRANSMIT DESTINATION = IUT, SOURCE = TD,

Read-Foreign-Device-Table

2. RECEIVE, DESTINATION = TD, SOURCE = IUT,

Read-Foreign-Device-Table-Ack,

List of FDT entries

3. CHECK (List of FDT entries is empty )

1. 执行永久外部设备注册

永久（无限期）外部设备注册没有规定，因此该测试已被删除。

1. 外部设备表定时器操作
2. 非零持续时间的外部设备表定时器操作

目的：验证IUT将运用FDT定时器操作：有限时间外部设备注册，重新注册，向所提供的Time-To-Live参数添加宽限期以及在计时器到期时清除FDT条目。

配置要求：TD应为外部设备FD2。IUT的FDT必须为空。

测试步骤：

1.TRANSMIT

DA = IUT,

SA = FD2,

Register-Foreign-Device,

'Time-To-Live' = 60

2.RECEIVE

DA = FD2,

SA = IUT,

BVLC-Result,

'Result Code' = 0

3. WAIT (10 seconds)

4. TRANSMIT

DA = IUT,

SA = FD2,

Read-Foreign-Device-Table

5. RECEIVE

DA = FD2,

SA = IUT,

Read-Foreign-Device-Table-Ack,

B/IP address of FD2, Time-To-Live = 60, Remaining-Time = 80

minus test execution time.

(50 is also acceptable if Protocol\_Revision < 7)

6.TRANSMIT

DA = IUT,

SA = FD2,

Register-Foreign-Device,

'Time-To-Live' = 40

7. RECEIVE

DA = FD2,

SA = IUT,

BVLC-Result,

'Result Code' = 0

8. WAIT (30 seconds)

9. TRANSMIT

DA = IUT,

SA = FD2,

Read-Foreign-Device-Table

10. RECEIVE

DA = FD2,

SA = IUT,

Read-Foreign-Device-Table-Ack,

B/IP address of FD2, Time-To-Live = 40, Remaining-Time = 40

minus test execution time

(10 is also acceptable if Protocol\_Revision < 7)

11. WAIT (50 seconds)

12. TRANSMIT

DA = IUT,

SA = FD2,

Read-Foreign-Device-Table

13. RECEIVE

DA = FD2,

SA = IUT,

Read-Foreign-Device-Table-Ack,

(No FDT entries)

注意事项：供应商应当说明FDT定时器的精度。

1. 零持续时间外部设备定时器操作

目的：验证IUT将利用Time-To-Live参数等于零以及在定时器结束时清除FDT条目进行外部设备注册。

配置要求：TD应是外部设备FD2。IUT的FDT必须为空。

1.TRANSMIT

DA = IUT,

SA = FD2,

Register-Foreign-Device,

'Time-To-Live' = 0

2. RECEIVE

DA = FD2,

SA = IUT,

BVLC-Result,

'Result Code' = 0

3. WAIT (10 seconds)

4. TRANSMIT

DA = IUT,

SA = FD2,

Read-Foreign-Device-Table

5. RECEIVE

DA = FD2,

SA = IUT,

Read-Foreign-Device-Table-Ack,

B/IP address of FD2,

Time-To-Live = 0,

Remaining-Time = 20 minus test execution time. (0 is also acceptable if Protocol\_Revision < 7)

6. WAIT (30 seconds)

7. TRANSMIT

DA = IUT,

SA = FD2,

Read-Foreign-Device-Table

8. RECEIVE

DA = FD2,

SA = IUT,

Read-Foreign-Device-Table-Ack,

(No FDT entries)

注意事项：供应商应当说明FDT定时器的精度。

1. 应该忽略的单播消息

目的：验证IUT将忽略Original-Unicast-NPDU消息。

测试步骤：

1.TRANSMIT DESTINATION = IUT, SOURCE = TD,

Original-Unicast-NPDU,

NPDU = Read-Property

2. CHECK (The IUT shall take no action )

1. 拒绝Delete-Foreign-Device-Table-Entry

目的：验证当提供的FDT条目无效时，ITU将处理Delete-Foreign-Device-Table-Entry消息。

配置要求：TD是外部设备FD1。ITU的FDT必须是空的。

测试步骤：

1. TRANSMIT

DA = IUT,

SA = FD1,

Register-Foreign-Device,

'Time-To-Live' = 120

2. RECEIVE

DA = FD1,

SA = IUT,

BVLC-Result,

'Result Code' = Successful completion

3. TRANSMIT

DA = IUT,

SA = FD1,

Read-Foreign-Device-Table

4. RECEIVE

DA = FD1,

SA = IUT,

Read-Foreign-Device-Table-Ack,

B/IP address of FD1, Time-To-Live = 120, Remaining-Time = ?

5. TRANSMIT

DA = IUT,

SA = FD1,

Delete-Foreign-Device-Table-Entry,

'FDT Entry' = FD2

6. RECEIVE

DA = FD1,

SA = IUT,

BVLC-Result,

'Result Code' = Delete-Foreign-Device-Table-Entry NAK

7. TRANSMIT

DA = IUT,

SA = FD1,

Read-Foreign-Device-Table

8. RECEIVE

DA = FD1,

SA = IUT,

Read-Foreign-Device-Table-Ack,

B/IP address of FD1, Time-To-Live = 120, Remaining-Time = ?

1. 执行Delete-Foreign-Device-Table-Entry

目的：验证当提供的FDT条目无效时，ITU将处理Delete-Foreign-Device-Table-Entry的消息。

配置要求：TD是外部设备FD1。ITU的FDT须是空的。

测试步骤：

1. TRANSMIT

DA = IUT,

SA = FD1,

Register-Foreign-Device,

'Time-To-Live' = 120

2. RECEIVE

DA = FD1,

SA = IUT,

BVLC-Result,

'Result Code' = Successful completion

3. TRANSMIT

DA = IUT,

SA = FD1,

Read-Foreign-Device-Table

4. RECEIVE

DA = FD1,

SA = IUT,

Read-Foreign-Device-Table-Ack,

B/IP address of FD1, Time-To-Live = 120, Remaining-Time = ?

5. TRANSMIT

DA = IUT,

SA = FD1,

Delete-Foreign-Device-Table-Entry,

'FDT Entry' = FD1

6. RECEIVE

DA = FD1,

SA = IUT,

BVLC-Result,

'Result Code' = Successful completion

7. TRANSMIT

DA = IUT,

SA = FD1,

Read-Foreign-Device-Table

8. RECEIVE

DA = FD1,

SA = IUT,

Read-Foreign-Device-Table-Ack,

No FDT Entries

* 1. 广播管理（BBMD，外部设备，本地应用程序）

这组测试验证IUT将执行广播分发的所有路径。

配置要求：IUT配置为BBMD选项开启，FDT选项开启。

FDT应包含以下两项：

|  |  |
| --- | --- |
| B/IP地址 | 生存时间 |
| FD1 | 3600 |
| FD2 | 3600 |

注意事项：每个外部设备注册的剩余时间必须满足能够在注册终止之前完成测试。

1. 直连IP子网的广播消息

目的：验证IUT正确地把Original-Broadcast-NPDU转发到其BDT中的IP子网、外部设备和本地应用程序。

测试思想：测试设备应广播Original-Broadcast-NPDU消息，如该设备是与IUT在同一IP子网上的节点一般。网络层报头中的DNET/DADR和SNET/SADR字段不存在，因为它们表示本地广播。

配置要求：TD应是设备D1。

该测试分为单跳分发测试和双跳分发测试。

1. 直连IP子网的广播消息（单跳分发）

配置要求：IUT的BDT 应该包含以下三个条目：

|  |  |
| --- | --- |
| B/IP 地址 | 广播分发掩码 |
| IUT | IP子网1 子网掩码 |
| BBMD1 | IP子网2 子网掩码 |
| BBMD2 | IP子网3 子网掩码 |

步骤2-5是向被认为是BAC net网络成员的设备分发Who-Is请求，

步骤6-10是从本地应用程序分发I-Am应答。

测试步骤：

1. TRANSMIT

DA = Local IP Broadcast,

SA = D1,

Original-Broadcast-NPDU,

NPDU = Who-Is

2. RECEIVE

DA = Directed IP Broadcast to IP Subnet 2,

SA = IUT,

Forwarded-NPDU,

Originating-Device = D1,

NPDU = Who-Is

3. RECEIVE

DA = Directed IP Broadcast to IP Subnet 3,

SA = IUT,

Forwarded-NPDU,

Originating-Device = D1,

NPDU = Who-Is

4. RECEIVE

DA = FD1,

SA = IUT,

Forwarded-NPDU,

Originating-Device = D1,

NPDU = Who-Is

5. RECEIVE

DA = FD2,

SA = IUT,

Forwarded-NPDU,

Originating-Device = D1,

NPDU = Who-Is

6. RECEIVE

DA = Local IP Broadcast,

SA = IUT,

Original-Broadcast-NPDU,

NPDU = I-Am

7. RECEIVE

DA = Directed IP Broadcast to IP Subnet 2,

SA = IUT,

Forwarded-NPDU,

Originating-Device = IUT,

NPDU = I-Am

8. RECEIVE

DA = Directed IP Broadcast to IP Subnet 3,

SA = IUT,

Forwarded-NPDU,

Originating-Device = IUT,

NPDU = I-Am

9. RECEIVE

DA = FD1,

SA = IUT,

Forwarded-NPDU,

Originating-Device = IUT,

NPDU = I-Am

10. RECEIVE

DA = FD2,

SA = IUT,

Forwarded-NPDU,

Originating-Device = IUT,

NPDU = I-Am

注意事项：IUT发送的消息的顺序并不重要。

1. 直连IP子网的广播消息(两跳分发)

配置要求：BDT应包含以下三项：

|  |  |
| --- | --- |
| B/IP 地址 | 广播分发掩码 |
| IUT | 255.255.255.255 |
| BBMD1 | 255.255.255.255 |
| BBMD2 | 255.255.255.255 |

步骤2-5是将Who-Is请求分发给被认为是BACnet网络成员的设备，步骤6-10是来自本地应用程序分发I-Am应答。

测试步骤：

1. TRANSMIT

DA = Local IP Broadcast,

SA = D1,

Original-Broadcast-NPDU,

NPDU = Who-Is

2. RECEIVE

DA = BBMD1,

SA = IUT,

Forwarded-NPDU,

Originating-Device = D1,

NPDU = Who-Is

3. RECEIVE

DA = BBMD2,

SA = IUT,

Forwarded-NPDU,

Originating-Device = D1,

NPDU = Who-Is

4. RECEIVE

DA = FD1,

SA = IUT,

Forwarded-NPDU,

Originating-Device = D1,

NPDU = Who-Is

5. RECEIVE

DA = FD2,

SA = IUT,

Forwarded-NPDU,

Originating-Device = D1,

NPDU = Who-Is

6. RECEIVE

DA = Local IP Broadcast,

SA = IUT,

Original-Broadcast-NPDU,

NPDU = I-Am

7. RECEIVE

DA = BBMD1,

SA = IUT,

Forwarded-NPDU,

Originating-Device = IUT,

NPDU = I-Am

8. RECEIVE

DA = BBMD2,

SA = IUT,

Forwarded-NPDU,

Originating-Device = IUT,

NPDU = I-Am

9. RECEIVE

DA = FD1,

SA = IUT,

Forwarded-NPDU,

Originating-Device = IUT,

NPDU = I-Am

10. RECEIVE

DA = FD2,

SA = IUT,

Forwarded-NPDU,

Originating-Device = IUT,

NPDU = I-Am

注意事项：IUT发送的消息的顺序并不重要。

1. 对等BBMD广播消息转发

目的：验证IUT将向本地网络、对等BBMD、外部设备和本地应用程序发送Forwarded-NPDU消息。

测试思想：TD应发送Forwarded-NPDU到IUT，就像它是对等的BBMD1一样。网络层报头中的DNET/DADR和SNET/SADR字段不出现。

配置要求：TD应该是BBMD1。

该测试分为单跳分发测试和双跳分发测试。

1. 对等BBMD广播消息转发(单跳分发)

配置要求：BDT应按照测试15.7.1.1配置。

步骤2-3是将Who-Is请求分发给被认为是BACnet网络成员的设备，步骤4-8是来自本地应用程序分发I-Am应答。

测试步骤：

1. TRANSMIT

DA = Directed IP Broadcast to IP Subnet 1,

SA = BBMD1,

Forwarded-NPDU,

Originating-Device = D2,

NPDU = Who-Is

2. RECEIVE

DA = FD1,

SA = IUT,

Forwarded-NPDU,

Originating-Device = D2,

NPDU = Who-Is

3. RECEIVE

DA = FD2,

SA = IUT,

Forwarded-NPDU,

Originating-Device = D2,

NPDU = Who-Is

4. RECEIVE

DA = Local IP Broadcast,

SA = IUT,

Original-Broadcast-NPDU,

NPDU = I-Am

5. RECEIVE

DA = Directed IP Broadcast to IP Subnet 2,

SA = IUT,

Forwarded-NPDU,

Originating-Device = IUT,

NPDU = I-Am

6. RECEIVE

DA = Directed IP Broadcast to IP Subnet 3,

SA = IUT,

Forwarded-NPDU,

Originating-Device = IUT,

NPDU = I-Am

7. RECEIVE

DA = FD1,

SA = IUT,

Forwarded-NPDU,

Originating-Device = IUT,

NPDU = I-Am

8. RECEIVE

DA = FD2,

SA = IUT,

Forwarded-NPDU,

Originating-Device = IUT,

NPDU = I-Am

注意事项：IUT发送的消息的顺序并不重要。

1. 对等BBMD广播消息转发(两跳分发)

配置要求：BDT应按照测试15.7.1.2配置。

步骤2-4是将Who-Is请求分发给被认为是BACnet网络成员的设备，步骤5-9是来自本地应用程序分发I-Am应答。

测试步骤：

1. TRANSMIT

DA = IUT,

SA = BBMD1,

Forwarded-NPDU,

Originating-Device = D2,

NPDU = Who-Is

2. RECEIVE

DA = Local IP Broadcast,

SA = IUT,

Forwarded-NPDU,

Originating-Device = D2,

NPDU = Who-Is

3. RECEIVE

DA = FD1,

SA = IUT,

Forwarded-NPDU,

Originating-Device = D2,

NPDU = Who-Is

4. RECEIVE

DA = FD2,

SA = IUT,

Forwarded-NPDU,

Originating-Device = D2,

NPDU = Who-Is

5. RECEIVE

DA = Local IP Broadcast,

SA = IUT,

Original-Broadcast-NPDU,

NPDU = I-Am

6. RECEIVE

DA = BBMD1,

SA = IUT,

Forwarded-NPDU,

Originating-Device = IUT,

NPDU = I-Am

7. RECEIVE

DA = BBMD2,

SA = IUT,

Forwarded-NPDU,

Originating-Device = IUT,

NPDU = I-Am

8. RECEIVE

DA = FD1,

SA = IUT,

Forwarded-NPDU,

Originating-Device = IUT,

NPDU = I-Am

9. RECEIVE

DA = FD2,

SA = IUT,

Forwarded-NPDU,

Originating-Device = IUT,

NPDU = I-Am

注意事项：IUT发送的消息的顺序并不重要。

1. 外部设备广播消息

目的：验证IUT将向本地网络、对等BBMD、外部设备和本地应用程序发送Forwarded-NPDU消息。

测试思想：TD将Distribute-Broadcast-To-Network发送给IUT，就像它是外部设备FD1一样。网络层报头中的DNET/DADR和SNET/SADR字段不出现。

配置要求：TD应该是FD1。

该测试分为单跳分发测试和双跳分发测试。

1. 外部设备广播消息（单跳分发）

配置要求：BDT应按照测试15.7.1.1配置。

步骤2-5是将Who-Is请求分发给被认为是BACnet网络成员的设备，步骤6-10是来自本地应用程序分发I-Am应答。

测试步骤：

1. TRANSMIT

DA = IUT,

SA = FD1,

Distribute-Broadcast-To-Network,

NPDU = Who-Is

2. RECEIVE

DA = Local IP Broadcast,

SA = IUT,

Forwarded-NPDU,

Originating-Device = FD1,

NPDU = Who-Is

3. RECEIVE

DA = Directed IP Broadcast to IP Subnet 2,

SA = IUT,

Forwarded-NPDU,

Originating-Device = FD1,

NPDU = Who-Is

4. RECEIVE

DA = Directed IP Broadcast to IP Subnet 3,

SA = IUT,

Forwarded-NPDU,

Originating-Device = FD1,

NPDU = Who-Is

5. RECEIVE

DA = FD2,

SA = IUT,

Forwarded-NPDU,

Originating-Device = FD1,

NPDU = Who-Is

6. RECEIVE

DA = Local IP Broadcast,

SA = IUT,

Original-Broadcast-NPDU,

NPDU = I-Am

7. RECEIVE

DA = Directed IP Broadcast to IP Subnet 2,

SA = IUT,

Forwarded-NPDU,

Originating-Device = IUT,

NPDU = I-Am

8. RECEIVE

DA = Directed IP Broadcast to IP Subnet 3,

SA = IUT,

Forwarded-NPDU,

Originating-Device = IUT,

NPDU = I-Am

9. RECEIVE

DA = FD1,

SA = IUT,

Forwarded-NPDU

Originating-Device = IUT,

NPDU = I-Am

10. RECEIVE

DA = FD2,

SA = IUT,

Originating-Device = IUT,

NPDU = I-Am

注意事项：IUT发送的消息的顺序并不重要。

1. 外部设备广播消息（双跳分发）

配置要求：BDT和FDI应按照测试15.7.1.2配置。

步骤2-5是将Who-Is请求分发给被认为是BACnet网络成员的设备，步骤6-10是来自本地应用程序分发I-Am应答。

测试步骤：

1. TRANSMIT

DA = IUT,

SA = FD1,

Distribute-Broadcast-To-Network,

NPDU = Who-Is

2. RECEIVE

DA = Local IP Broadcast,

SA = IUT,

Forwarded-NPDU,

Originating-Device = FD1,

NPDU = Who-Is

3. RECEIVE

DA = BBMD1,

SA = IUT,

Forwarded-NPDU,

Originating-Device = FD1,

NPDU = Who-Is

4. RECEIVE

DA = BBMD2,

SA = IUT,

Forwarded-NPDU,

Originating-Device = FD1,

NPDU = Who-Is

5. RECEIVE

DA = FD2,

SA = IUT,

Forwarded-NPDU,

Originating-Device = FD1,

NPDU = Who-Is

6. RECEIVE

DA = Local IP Broadcast,

SA = IUT,

Original-Broadcast-NPDU,

NPDU = I-Am

7. RECEIVE

DA = BBMD1,

SA = IUT,

Forwarded-NPDU,

Originating-Device = IUT,

NPDU = I-Am

8. RECEIVE

DA = BBMD2,

SA = IUT,

Forwarded-NPDU,

Originating-Device = IUT,

NPDU = I-Am

9. RECEIVE

DA = FD1,

SA = IUT,

Forwarded-NPDU,

Originating-Device = IUT,

NPDU = I-Am

10. RECEIVE

DA = FD2,

SA = IUT

Forwarded-NPDU,

Originating-Device = IUT,

NPDU = I-Am

注意事项：IUT发送的消息的顺序并不重要。

* 1. 注册为外部设备

所需测试：无

BACnet参考条款：J.5.2

目的：此测试用例验证了IUT可以注册为具有BBMD的外部设备。

测试思想：将IUT注册为具有TD的外部设备。

配置要求：配置IUT使其注册为具有TD的外部设备。

测试步骤：

1. RECEIVE DESTINATION = TD, SOURCE = IUT,

Register-Foreign-Device

2. TRANSMIT DESTINATION = IUT, SOURCE = TD,

BVLC-Result,

‘Result Code’ = Successful completion

* 1. 初始化包含NPDU的BVLL服务请求

这组测试验证IUT可以正确初始化包含NPDU的BVLL服务请求。

1. 分布式广播到网络

所需测试：15.X1，“注册为外部设备”

BACnet 参考条款：J.2.10

目的：验证注册为外部设备的IUT可以向BBMD发出请求，以便在BBMD的BDT中的所有子网上广播消息。

测试思想：将IUT配置为注册自己成具有TD的外部设备，然后在完成注册后，发起广播消息传送到BBMD进行分发。如果IUT不支持外部设备注册，或者不能发起广播消息来传输BACnet NPDU，那么该测试将被省略。

测试步骤：

1. RECEIVE DESTINATION = TD, SOURCE = IUT,

Register-Foreign-Device

2. TRANSMIT DESTINATION = IUT, SOURCE = TD,

BVLC-Result,

‘Result Code’ = Successful completion

3. MAKE (the IUT initiate a broadcast)

4. RECEIVE DESTINATION = TD, SOURCE = IUT,

Distribute-Broadcast-To-Network

1. 发起原单播NPDU

所需测试：15.2.3，“执行Original-Unicast-NPDU”

BACnet参考条款：J.2.11

目的：验证IUT可以向另一个设备发出一个定向NPDU。

测试思想：TD在Original-Unicast-NPDU中向IUT发送ReadProperty-Request。IUT在Original-Unicast-NPDU中用 ReadProperty-ACK应答。

测试步骤：

1. TRANSMIT

DA = IUT,

SA = D1,

Original-Unicast-NPDU,

NPDU = Read-Property

2. RECEIVE

DA = D1,

SA = IUT,

Original-Unicast-NPDU,

NPDU = Read-Property-Ack

1. 原始广播NPDU

BACnet参考条款：J.2.12

目的：验证IUT可以在它自己的IP子网上发布广播。

测试思想：使IUT在其IP子网上发起广播消息。如果IUT不能发起广播消息来传输BACnet NPDU，则省略该测试。

测试步骤：

1. MAKE (the IUT initiate a broadcast)

2. RECEIVE DA = Local IP Broadcast, SOURCE = IUT,

Original-Broadcast-NPDU

1. 报告测试结果

向供应商提供测试结果的报告，其中包含：

(a)执行的每个测试用例的总结；

(b)执行的每个测试用例的结果(通过/失败)；

(c)对于失败的测试用例可获得的任何诊断信息。

附 录 A

（资料性附录）

电子协议实现一致性声明(EPICS) 示例

A1 电子协议实现一致性声明(EPICS) 示例（信息）

（此附录不作为标准的一部分，仅供信息参考）

PIC0

BACnet协议实现一致性声明

--本EPICS文件样例说明了在条款4中定义的格式。

--包含了至少每种对象类型中的一种，并用此作为模版。

--为特殊的BACnet设备创建一个EPICS文件。

供应商名称：“ASHRAE”

产品名称：“标准BACnet设备”

产品型号：“1.0”

产品描述：“一个很非常棒的东西！”

--删除任何不支持的BIBBS。

BIBBs Supported:

{

DS-RP-A

DS-RP-B

DS-RPM-A

DS-RPM-B

DS-RPC-A

DS-RPC-B

DS-WP-A

DS-WP-B

DS-WPM-A

DS-WPM-B

DS-COV-A

DS-COV-B

DS-COVP-A

DS-COVP-B

DS-COVU-A

DS-COVU-B

AE-N-A

AE-N-I-B

AE-N-E-B

AE-ACK-A

AE-ACK-B

AE-ASUM-A

AE-ASUM-B

AE-ESUM-A

AE-ESUM-B

AE-INFO-A

AE-INFO-B

AE-LS-A

AE-LS-B

SCHED-A

SCHED-I-B

SCHED-E-B

T-VMT-A

T-VMT-I-B

T-VMT-E-B

T-ATR-A

T-ATR-B

DM-DDB-A

DM-DDB-B

DM-DOB-A

DM-DOB-B

DM-DCC-A

DM-DCC-B

DM-PT-A

DM-PT-B

DM-TM-A

DM-TM-B

DM-TS-A

DM-TS-B

DM-UTC-A

DM-UTC-B

DM-RD-A

DM-RD-B

DM-BR-A

DM-BR-B

DM-R-A

DM-R-B

DM-LM-A

DM-LM-B

DM-OCD-A

DM-OCD-B

DM-VT-A

DM-VT-B

NM-CE-A

NM-CE-B

NM-RC-A

NM-RC-B

}

--删除完全不支持的所有服务。

--酌情移除Initiate（发起）和Execute（执行）。

BACnet Standard Application Services Supported:

{

AcknowledgeAlarm Initiate Execute

ConfirmedCOVNotification Initiate Execute

UnconfirmedCOVNotification Initiate Execute

ConfirmedEventNotification Initiate Execute

UnconfirmedEventNotification Initiate Execute

GetAlarmSummary Initiate Execute

GetEnrollmentSummary Initiate Execute

GetEventInformation Initiate Execute

LifeSafetyOperation Initiate Execute

SubscribeCOV Initiate Execute

SubscribeCOVProperty Initiate Execute

AtomicReadFile Initiate Execute

AtomicWriteFile Initiate Execute

AddListElement Initiate Execute

RemoveListElement Initiate Execute

CreateObject Initiate Execute

DeleteObject Initiate Execute

ReadProperty Initiate Execute

ReadPropertyConditional Initiate Execute

ReadPropertyMultiple Initiate Execute

ReadRange Initiate Execute

WriteProperty Initiate Execute

WritePropertyMultiple Initiate Execute

DeviceCommunicationControl Initiate Execute

ConfirmedPrivateTransfer Initiate Execute

UnconfirmedPrivateTransfer Initiate Execute

ReinitializeDevice Initiate Execute

ConfirmedTextMessage Initiate Execute

UnconfirmedTextMessage Initiate Execute

TimeSynchronization Initiate Execute

UTCTimeSynchronization Initiate Execute

Who-Has Initiate Execute

I-Have Initiate Execute

Who-Is Initiate Execute

I-Am Initiate Execute

VT-Open Initiate Execute

VT-Close Initiate Execute

VT-Data Initiate Execute

RequestKey Initiate Execute

Authenticate Initiate Execute

}

--删除任何完全不支持的对象类型。

--酌情移除Creatable（可创建）或Deleteable（可删除）。

Standard Object Types Supported:

{

Accumulator Creatable Deleteable

Analog Input Creatable Deleteable

Analog Output Creatable Deleteable

Analog Value Creatable Deleteable

Averaging Creatable Deleteable

Binary Input Creatable Deleteable

Binary Output Creatable Deleteable

Binary Value Creatable Deleteable

Calendar Creatable Deleteable

Command Creatable Deleteable

Device Creatable Deleteable

Event Enrollment Creatable Deleteable

File Creatable Deleteable

Group Creatable Deleteable

Life Safety Point Creatable Deleteable

Life Safety Zone Creatable Deleteable

Loop Creatable Deleteable

Multi-state Input Creatable Deleteable

Multi-state Output Creatable Deleteable

Multi-state Value Creatable Deleteable

Notification Class Creatable Deleteable

Program Creatable Deleteable

Pulse Converter Creatable Deleteable

Schedule Creatable Deleteable

Trend Log Creatable Deleteable

}

--移除所有不支持的DL选项。

Data Link Layer Option:

{

ISO 8802-3, 10BASE5

ISO 8802-3, 10BASE2

ISO 8802-3, 10BASET

ISO 8802-3, Fiber

ARCNET, coax star

ARCNET, coax bus

ARCNET, twisted pair star

ARCNET, twisted pair bus

ARCNET, fiber star

MS/TP master. Baud rate(s): 9600

MS/TP slave. Baud rate(s): 9600

Point-To-Point. EIA 232, Baud rate(s)

Point-To-Point. Modem, Baud rate(s): 14.4k

Point-To-Point. Modem, Autobaud range: 9600 to 28.8k

BACnet/IP, “DIX” Ethernet

BACnet/IP, Other

LonTalk

Other

}

--移除所有不支持的字符集。

Character Sets Supported:

{

ANSI X3.4

IBM/Microsoft DBCS

ISO 8859-1

JIS C 6226

ISO 10646 (UCS-4)

ISO 10646 (UCS-2)

}

--用合适的值替换APDU段数的最大值。

--酌情移除段落分割线或改变窗口大小。

--当设备不是路由器时移除Router。

--当此功能不被支持时移除BACnet/IP BBMD。

Special Functionality:

{

Maximum APDU size in octets: 1476

Segmented Requests Supported, window size: 3

Segmented Responses Supported, window size: 3

Router

BACnet/IP BBMD

}

--仅包含适用的约束。适当调整参数。

--覆盖出现异常情况的对象数据库的约束条件。

--空列表表明没有定义全局约束。

Default Property Value Restrictions:

{

unsigned-integer: <minimum 0; maximum 65535>

signed integer: <minimum ??; maximum ??>

real <minimum ??; maximum ??; resolution ??>

double <minimum ??; maximum ??; resolution ??>

date: <minimum ??; maximum ??>

octet-string: <maximum length string 512>

character string: <maximum length string 128>

list: <maximum length list 10>

variable-length array: <maximum length array10>

}

--移除所有不在数据库中的对象类型。

--按需要复制对象类型模版来获取数据库中的每一个对象的类型。

--替换属性以指示数据库中属性的实际值。

--如果该值决于传感器的输入而不能确定，则用“？”表示该值。

--所有可写的属性值后都应当加上一个W。

List of Objects in test device:

{

{

object-identifier: (accumulator, 1)

object-name: "meter 1"

object-type: ACCUMULATOR

present-value: 125

description: ""

device-type: "electric pulse"

status-flags: {FALSE,FALSE,FALSE,FALSE}

event-state: NORMAL

reliability: NO-FAULT-DETECTED

out-of-service: FALSE

scale: 0

units: KILOWATT\_HOURS

prescale: (1,10000)

max-pres-value: 9999

value-change-time: {(Tuesday, 2-January-2007), 9:39:21.02}

value-before-change: 0

value-set: 67 W

logging-record: {{(Tuesday, 2-January-2007), 9:39:33.01}, 0, 27,NORMAL}

logging-object: (trend-log, 3)

pulse-rate: 3

high-limit: 15

low-limit: 0

limit-monitoring-interval: 300

notification-class: 1

time-delay: 10

limit-enable: {TRUE, TRUE}

event-enable: {TRUE, FALSE, TRUE}

acked-transitions: {TRUE, TRUE, TRUE}

notify-type: ALARM

event-time-stamps: {{(Monday, 1-January-2007),18:50:21.02},

{(\*-\*-\*),\*:\*:\*.\*}, {(Monday, 1-January-2007), 18:51:34.0}}

profile-name: ""

},

{

object-identifier: (analog-input, 1)

object-name: "1AH1MAT" W

object-type: ANALOG\_INPUT

present-value: 58.1

description: "Mixed Air Temperature"

device-type: "1000 Ohm RTD"

status-flags: {FALSE,FALSE,FALSE,FALSE}

event-state: NORMAL

reliability: NO-FAULT-DETECTED

out-of-service: FALSE

update-interval: 10

units: DEGREES-FAHRENHEIT

min-pres-value: -50.0

max-pres-value: 250.0

resolution: 0.1

COV-increment: 0.2

time-delay: 10

notification-class: 3

high-limit: 60.0

low-limit: 55.0

deadband: 1.0

limit-enable: {TRUE, TRUE}

event-enable: {TRUE, FALSE, TRUE}

acked-transitions: {TRUE, TRUE, TRUE}

notify-type: EVENT

event-time-stamps:{{(Monday,24-January-1998),18:50:21.02},{(\*-\*-\*),\*:\*:\*.\*},{(23-March-1998), 19:01:34.0}}

},

{

object-identifier: (analog-output, 1)

object-name: "1AH1DMPR" W

object-type: ANALOG\_OUTPUT

present-value: 75.0

description: "Damper Actuator"

device-type: "3-8 PSI Actuator"

status-flags: {FALSE,FALSE,FALSE,FALSE}

event-state: NORMAL

reliability: NO-FAULT-DETECTED

out-of-service: FALSE

units: PERCENT

min-pres-value: 0.0

max-pres-value: 100.0

resolution: 0.1

priority-array: {?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?} R

relinquish-default: 50.0

},

{

object-identifier: (analog-value, 1)

object-name: "1AH1ENTH" W

object-type: ANALOG\_VALUE

present-value: 38.1

description: "Enthalpy"

status-flags: {FALSE,FALSE,FALSE,FALSE}

event-state: NORMAL

reliability: NO-FAULT-DETECTED

out-of-service: FALSE

units: BTUS\_PER\_POUND\_DRY\_AIR

priority-array: {?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?} R

relinquish-default: 50.0

COV-increment: 0.2

time-delay: 10

notification-class: 3

high-limit: 60.0

low-limit: 20.0

deadband: 1.0

limit-enable: {TRUE, TRUE}

event-enable: {TRUE, FALSE, TRUE}

acked-transitions: {TRUE, TRUE, TRUE}

notify-type: EVENT

event-time-stamps: {6,3,5}

},

{

object-identifier: (averaging, 1)

object-name: "FLR 12 DEMAND"

object-type: AVERAGING

minimum-value: 2.4

minimum-value-timestamp: (16-December-1999,13:15:07.32)

average-value: 12.7

maximum-value: 18.8

maximum-value\_Timestamp: (16-December-1999,13:06:12.19)

description: "Floor 12 Electrical Demand"

attempted-samples: 15

valid-samples: 14

object-property-reference: (analog-input, 12)

window-interval: 900

window-samples: 15

},

{

object-identifier: (binary-input, 1)

object-name: "HighPressSwitch"

object-type: BINARY\_INPUT

present-value: ACTIVE

description: "Penthouse Supply High Static"

device-type: "ABC Pressure Switch"

status-flags: {TRUE,FALSE,FALSE,FALSE}

event-state: OFFNORMAL

reliability: NO-FAULT-DETECTED

out-of-service: FALSE

polarity: NORMAL

inactive-text: "Static Pressure OK"

active-text: "High Pressure Alarm"

change-of-state-time: {(23-March-1995),19:01:34.0}

change-of-state-count: 134

time-of-state-count-reset: {(1-January-1995),00:00:00.0}

elapsed-active-time: 401

time-of-active-time-reset: {(1-January-1995),00:00:00.0}

time-delay: 10

notification-class: 3

alarm-value: ACTIVE

event-enable: {TRUE, FALSE, TRUE}

acked-transitions: {FALSE, TRUE, TRUE}

notify-type: ALARM

event-time-stamps: {09:02:31.05,14:07:21.57,13:33:02.82}

},

{

object-identifier: (binary-output, 1)

object-name: "Floor3ExhaustFan"

object-type: BINARY\_OUTPUT

present-value: INACTIVE

description: "Third floor bathroom exhaust fan"

device-type: "ABC 100 Relay"

status-flags: {FALSE,FALSE,FALSE,FALSE}

event-state: NORMAL

reliability: NO\_FAULT\_DETECTED

out-of-service: FALSE

polarity: REVERSE

inactive-text: "Fan is turned off"

active-text: "Fan is running"

change-of-state-time: {(23-March-1995),19:01:34.0}

change-of-state-count: 47

time-of-state-count-reset: {(1-January-1995),00:00:00.0}

elapsed-active-time: 650

time-of-active-time-reset: {(1-January-1995),00:00:00.0}

minimum-off-time: 100

minimum-on-time: 10

priority-array: {NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, INACTIVE}

relinquish-default: INACTIVE

time-delay: 10

notification-class: 3

feedback-value: ACTIVE

event-enable: {TRUE, FALSE, TRUE} acked-transitions: {FALSE, TRUE, TRUE}

notify-type: EVENT

event-time-stamps: {{(Monday,24-January-1998),18:50:21.02},{(\*-\*-\*),\*:\*:\*.\*},{(23-March-1998), 19:01:34.0}}

},

{

object-identifier: (binary-value, 1)

object-name: "ExhaustFanEnable"

object-type: BINARY\_VALUE

present-value: ACTIVE

description: "Exhaust Fan Operator"

status-flags: {FALSE,FALSE,FALSE,FALSE}

event-state: NORMAL

reliability: NO\_FAULT\_DETECTED

out-of-service: FALSE

inactive-text: "Enabled by Operator"

active-text: "Fan Not Enabled by Operator" change-of-state-time: {(23-March-1995),19:01:34.0}

change-of-state-count: 134

time-of-state-count-reset: {(1-January-1995),00:00:00.0}

elapsed-active-time: 401

time-of-active-time-reset: {(1-January-1995),00:00:00.0}

minimum-off-time: 0

minimum-on-time: 0

priority-array: {NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, ACTIVE}

relinquish-default: INACTIVE

time-delay: 10

notification-class: 3

alarm-value: ACTIVE

event-enable: {TRUE, FALSE, TRUE}

acked-transitions: {FALSE, TRUE, TRUE}

notify-type: EVENT

event-time-stamps: {{(Monday,24-January-1998),18:50:21.02},{(\*-\*-\*),\*:\*:\*.\*},{(23-March-1998), 19:01:34.0}}

},

{

object-identifier: (calendar, 1)

object-name: "HOLIDAYS"

object-type: CALENDAR

description: "1995 School District Holidays"

present-value: TRUE

date-list: ((19-February-1995),(28-May-1995),{(24-December-1995)-(4-January-1996)})

},

{

object-identifier: (Command, 1)

object-name: "ZONE43CONTROL"

object-type: COMMAND

description: "Fourth Floor, West Wing Office Suite"

present-value: 1

in-process: FALSE

all-writes-successful: TRUE

action: {({,(analog-value, 5),present-value,,65.0,,,TRUE,TRUE},{,(binary-output, 3),present-value,,

INACTIVE,8,1,TRUE,TRUE}),

({,(analog-value, 5), present-value,,72.0,,,TRUE,TRUE},{,(binary-output, 3),

present-value,, ACTIVE,8,2,TRUE,TRUE})

}

action-text: {"Unoccupied", "Occupied"}

},

{

object-identifier: (device, 90)

object-name: "AC1 System Controller" W

object-type: DEVICE

system-status: OPERATIONAL

vendor-name: "ABC Controls"

vendor-identifier: 0

model-name: "1000 Plus"

firmware-revision: "1.2 "

application-software-version: "V4.0 - April 12, 2005"

location: "Basement Mechanical Room"

protocol-version: 1

protocol-revision: 4

protocol-services-supported:

{

T, T, T, T, T, T, -- Alarm and event

T, T, -- File

T, T, T, T, T, T, -- Create, Delete, Read

T, T, T, -- ReadMultiple, Write, WriteMultiple

T, T, T, T, -- cPrivateTransfer, Re-init

T, T, T, -- VT Open, Data, Close

T, T, -- Security

T, T,T, T, T, -- I-Am, I-Have, uPrivateTransfer

T, T, T, T, -- TimeSync, Who-Has, Who-Is

T, T, T, T, -- ReadRange, UTCTimeSync, LifeSafetyOperation, SubscribeCOVProperty

T -- GetEventInfo

}

protocol-object-types-supported:

{

T, T, T, T, T, T, -- AI, AO, AV,BI, BO, BV

T, T, T, T, T, T, -- calendar, command, device, event enrollment, file, group

T, T, T, T, T, T, -- loop, MSI, MSO, notification class, program, schedule

T, T, T, T, T, -- averaging, multi-state-value, trend-log, life-safety-point, life-safety-zone

T, T -- accumulator, pulse-converter

}

object-list:

{

(accumulator, 1),

(analog-input, 1),

(analog-output, 1),

(analog-value, 1),

(averaging, 1),

(binary-input, 1),

(binary-output, 1),

(binary-value, 1),

(calendar, 1),

(command, 1),

(device, 90),

(event-enrollment, 1),

(file, 1),

(group, 1),

(life-safety-point, 1),

(life-safety-zone),

(loop, 1),

(multi-state-input, 1),

(multi-state-output, 1),

(multi-state-value, 1)

(notification-class, 1),

(program, 1),

(pulse-converter, 1),

(schedule, 1),

(trend-log, 1),

}

max-APDU-length-accepted: 1476

segmentation-supported: segmented-both

vt-classes-supported: (DEFAULT-TERMINAL, DEC-VT100)

local-time:?

local-date: ?

utc-offset: 6.0

daylight-savings-status: FALSE

apdu-segment-timeout: 2000

apdu-timeout: 4000

number-of-APDU-retries: 3

list-of-session-keys: ({X”3799246237984589”,{1, X”3”}}, {X”4446214686489744”,{ 1, X”5”}})

--device-address-binding: () --empty list example

device-address-binding: ({(Device,1), {1, X”1”}},

{(Device,12),{1, X”23”}},

{(Device,40),{2, X”02608C41A606”}})

database-revision: 53

configuration-files: {(file, 1),(file, 2),(file, 3),}

last-restore-time: {(29-September-1989),01:00:00.00}

backup-failure-timeout: 300

active-cov-subscriptions:({{(device,12),300},{(analog-input,1), present-value},TRUE,100,1.0},

({(device, 40), 600},{(analog-input, 1),present-value},TRUE,3,1.5})

},

{

object-identifier: (event-enrollment, 1)

object-name: "Zone1\_Alarm"

object-type: EVENT\_ENROLLMENT

description: "Zone 1 Alarms"

event-type: OUT\_OF\_RANGE

notify-type: ALARM

event-parameters: {30, 65.0, 85.0, 0.25}

object-property-reference: {(analog-input, 2), present-value}

event-state: HIGH\_LIMIT

event-enable: {TRUE, TRUE, TRUE}

acked-transitions: {FALSE, TRUE, TRUE}

notification-class: 1

},

{

object-identifier: (file, 7)

object-name: "TREND\_AI1"

object-type: FILE

description: "Trend of AI1"

file-type: "TREND"

file-size: 45

modification-date: {(1-November-1995),08:30:49.0}

archive: FALSE

read-only: FALSE

file-access-method: RECORD\_ACCESS

},

{

object-identifier: (Group, 1)

object-name: "ZONE1\_TEMPS"

object-type: GROUP

description: "Zone 1 Temperature Group"

list-of-group-members: (

{(analog-input, 8),(present-value, reliability, description)},

{(analog-input, 9),(present-value, reliability, description)},

{(analog-input, 10),(present-value, reliability, description)},

{(analog-input, 11),(present-value, reliability, description)},

{(analog-input, 12),(present-value, reliability, description)}

)

present-value: ?

},

{

object-identifier: (life-safety-point, 2)

object-name: "SMK3W"

object-type: LIFE\_SAFETY\_POINT

present-value: PREALARM

tracking-value: PREALARM

description: "Floor 3, West Zone Smoke Detector"

device-type: "Old Smokey model 123"

status-flags: {TRUE, FALSE, FALSE, FALSE}

event-state: LIFE\_SAFETY\_ALARM

reliability: NO\_FAULT\_DETECTED

out-of-service: FALSE

mode: ON

time-delay: 10

notification-class: 39

life-safety-alarm-values: (ALARM)

alarm-values: (PREALARM)

fault-values: (FAULT)

event-enable: {TRUE, TRUE, TRUE}

acked-transitions: {TRUE, TRUE, TRUE}

notify-type: ALARM

event-time-stamps: {{(23-March-1995), 18:50:21.2},{(\*-\*-\*), \*:\*:\*.\*}, {(23-March-1995),

19:01:34.0)}}

silenced: SILENCE\_AUDIBLE

operation-expected: RESET\_ALARM

maintenance-required: NONE

setting: 50

direct-reading: 84.3

units: PERCENT\_OBSCURATION\_PER\_METER

member-of: ((life-safety-zone, 5))

},

{

object-identifier: (life-safety-zone, 2)

object-name: "SMK3"

object-type: LIFE\_SAFETY\_ZONE

present-value: PREALARM

tracking-value: PREALARM

description: "Floor 3 Smoke"

status-flags: {TRUE FALSE, FALSE, FALSE}

event-state: LIFE\_SAFETY\_ALARM

reliability: NO\_FAULT\_DETECTED

out-of-service: FALSE

mode: ON

time-delay: 10

notification-class: 39

life-safety-alarm-values: (ALARM)

alarm-values: (PREALARM)

fault-values: (FAULT)

event-enable: {TRUE, TRUE, TRUE}

acked-transitions: {TRUE, TRUE, TRUE}

notify-type: ALARM

event-time-stamps: {{(23-March-1995), 18:50:21.2},{(\*-\*-\*), \*:\*:\*.\*}, {(23-March-1995),

19:01:34.0)}}

silenced: UNSILENCED

operation-expected: SILENCE\_AUDIBLE

maintenance-required: NONE

zone-members: ((life-safety-point, 22),(life-safety-point, 23))

member-of: ((life-safety-zone, 5))

},

{

object-identifier: (loop, 1)

object-name: "AHU\_SAT\_LOOP"

object-type: LOOP

present-value: 8.3

description: "Supply air temp. PI control"

status-flags: {FALSE,FALSE,FALSE,FALSE}

event-state: NORMAL

reliability: NO\_FAULT\_DETECTED

out-of-service: FALSE

update-interval: 1

output-units: POUNDS-FORCE-PER-SQUARE-INCH

manipulated-variable-reference: {(analog-output, 5),present-value}

controlled-variable-reference: {(analog-input, 3),present-value}

controlled-variable-value: 56.1

controlled-variable-units: DEGREES-FAHRENHEIT

setpoint-reference: {(analog-value, 7),present-value}

setpoint: 57.0

action: DIRECT

proportional-constant: 0.5

proportional-constant-units: PSI\_PER\_DEGREE\_FAHRENHEIT

integral-constant: 0.1

integral-constant-units: PER\_MINUTE

derivative-constant: 0.0

derivative-constant-units: NO\_UNITS

bias: 9.0

maximum-output: 15.0

minimum-output: 3.0

priority-for-writing: 10

COV-increment: 0.2

time-delay: 3

notification-class: 1

error-limit: 5.0

event-enable: {TRUE, TRUE, TRUE}

acked-transitions: {TRUE, TRUE, TRUE}

notify-type: ALARM

},

{

object-identifier: (multi-state-input, 1)

object-name: "Fan1\_Input"

object-type: MULTI\_STATE\_INPUT

present-value: 2

description: "2-speed Fan#1"

device-type: "ZZZ Fan Motor"

status-flags: {FALSE,FALSE,FALSE,FALSE}

event-state: NORMAL

reliability: NO\_FAULT\_DETECTED

out-of-service: FALSE

number-of-states: 3

state-text: {"Off","On\_Low","On\_High"}

time-delay: 3

notification-class: 4

alarm-values: (3)

fault-values: (2)

event-enable: {TRUE, TRUE, TRUE}

acked-transitions: {TRUE, TRUE, TRUE}

notify-type: EVENT

},

{

object-identifier: (multi-state-output,1)

object-name: "Fan1\_Output"

object-type: MULTI\_STATE\_OUTPUT

present-value: 2

description: "2-speed Fan#1"

device-type: "ABC Fan Model A-6"

status-flags: {FALSE,FALSE,FALSE,FALSE}

event-state: OFFNORMAL

reliability: NO\_FAULT\_DETECTED

out-of-service: FALSE

number-of-states: 3

state-text: {"Off","On\_Low","On\_High"}

priority-array:{NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, 2, NULL, NULL, NULL,

NULL, NULL, NULL, NULL}

relinquish-default: 1

time-delay: 3

notification-class: 4

feedback-value: 3

event-enable: {TRUE, TRUE, TRUE}

acked-transitions: {TRUE, TRUE, TRUE}

notify-type: EVENT

},

{

object-identifier: (multi-state-value, 1)

object-name: "Control Mode"

object-type: MULTI\_STATE\_VALUE

present-value: 2

description: "Output of control mode algorithm"

status-flags: {FALSE,FALSE,FALSE,FALSE}

event-state: NORMAL

reliability: NO\_FAULT\_DETECTED

out-of-service: FALSE

number-of-states: 3

state-text: {"Heating", "Economizer Cooling", "Mechanical Cooling"}

},

{

object-identifier: (notification-class, 1)

object-name: "Alarms1"

object-type: NOTIFICATION\_CLASS

description: "Critical System Alarms"

notification-class: 1

priority: {3, 10, 10}

ack-required: {TRUE, TRUE, TRUE}

recipient-list: ( {{Monday, Tuesday, Wednesday, Thursday, Friday}, 6:00, 20:00,

(device, 12), 21, TRUE, {TRUE, TRUE,TRUE}},

{{Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday}, 0:00, 6:00,

(device, 18), 5, TRUE, {TRUE, TRUE, FALSE}},

{{Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday},

20:00, 24:00,(device, 18), 5, TRUE, (TRUE, TRUE, FALSE)}

)

},

{

object-identifier: (Program, 1)

object-name: "SomeAverage"

object-type: PROGRAM

program-state: RUNNING

program-change: READY

reason-for-halt: NORMAL

description-of-halt: "Normal"

program-location: "Line 2"

description: "Average of Somethings"

instance-of: "ThreeWayAverager"

status-flags: {FALSE,FALSE,FALSE,FALSE}

reliability: NO\_FAULT\_DETECTED

out-of-service: FALSE

},

{

object-identifier: (pulse-converter, 1)

object-name: "Meter 5"

object-type: PULSE\_CONVERTER

description: ""

present-value: 125.0

input-reference: {(accumulator, 1), present-value}

status-flags: {FALSE,FALSE,FALSE,FALSE}

event-state: NORMAL

reliability: NO-FAULT-DETECTED

out-of-service: FALSE

units: LITERS\_PER\_HOUR

scale-factor: 0.5

adjust-value: 500.0

count: 250

update-time: {(Thursday, 4-January-2007), 9:39:21.02}

count-change-time: {(Thursday, 4-January-2007), 9:39:41.52}

count-before-change: 523

cov-increment: 10.0

cov-period: 3600

notification-class: 1

time-delay: 0

high-limit: 1000.0

low-limit: 0.0

deadband: 0.0

limit-enable: {FALSE, TRUE}

event-enable: {TRUE, FALSE, TRUE}

acked-transitions: {TRUE, TRUE, TRUE}

notify-type: ALARM

event-time-stamps: {{(Monday, 1-January-2007),18:50:21.02},

{(\*-\*-\*),\*:\*:\*.\*},{(Monday, 1-January-2007), 18:51:34.0}}

profile-name: ""

},

{

object-identifier: (schedule, 2)

object-name: "Rm208Sched"

object-type: SCHEDULE

present-value: ACTIVE

description: "Room 208 Schedule"

effective-period: {(5-September-1995)-(10-June-1996)}

weekly-schedule:{({8:00,ACTIVE},{17:00,INACTIVE}),

({8:00,ACTIVE}),

({8:00,ACTIVE},{17:00,INACTIVE}),

({8:00,ACTIVE},{17:00,INACTIVE},{19:00,ACTIVE},{23:30,INACTIVE}),

({8:00,ACTIVE},{17:00,INACTIVE}),

({00:00,INACTIVE{),

({10:00,ACTIVE},{17:00,INACTIVE})

}

exception-schedule: {{(23-November-1995),({0:00,INACTIVE}),10},

{(calendar,1),({0:00,INACTIVE}),11},

{{(5-March-1996)-(7-March-1996)},({9:00,ACTIVE},{14:00,INACTIVE}),6}

}

schedule-default: INACTIVE

list-of-object-property-references: ((binary-output, 9),present-value)

priority-for-writing: 15

reliability: NO\_FAULT\_DETECTED

out-of-service: FALSE

profile-name: ""

},

{

object-identifier: (trend-log, 1)

object-name: "Room 3 Log"

object-type: TREND\_LOG

description: "Room 3 Temperature"

log-enable: TRUE

log-deviceobjectproperty: {(device, 100), (analog input, 3), present-value}

log-interval: 6000

stop-when-full: FALSE

buffer-size: 250

log-buffer: ()

record-count: 250

total-record-count: 131040

notification-threshold: 83

records-since-notification: 30

last-notify-record: 131010

event-state: NORMAL

notification-class: 1

event-enable: {FALSE, TRUE, TRUE}

acked-transitions: {TRUE, TRUE, TRUE}

notify-type: EVENT

event-time-stamps: {{(Monday, 1-January-2007),18:50:21.02},

{(\*-\*-\*),\*:\*:\*.\*},{(Monday, 1-January-2007), 18:51:34.0}}

profile-name: ""

}

}

End of BACnet Protocol Implementation Conformance Statement

--（BACnet协议实现一致性声明结束）

参 考 文 献

1. EIA/CEA-709.1-B（2002） Control network protocol specification
2. FIPS 46-2 (1993) Federal information processing standards—Data encryption standard
3. FIPS 180-2 (2002) Federal information processing standards—Secure hash standard
4. FIPS 197 (2002) Federal information processing standards—Advanced encryption standard
5. FIPS 198a（2002）The Keyed-hash message authentication code
6. IEEE Standard 754（2008） IEEE standard for binary floating-point arithmetic
7. IETF RFC 1321（1999） The MD5 message—Digest algorithm
8. IETF RFC 2246 (1999) The TLS protocol Version 1.0
9. IETF RFC 2616 (1999) Hypertext transfer protocol—HTTP/1.1
10. IETF RFC 2617 (1999) HTTP authentication: Basic and digest access authentication
11. INCITS 92 (1981,R1998) Data encryption algorithm
12. INCITS X3.4 (1986,R1997) Information processing—Coded character sets—7-bit American national standard code for information interchange (7-bit ASCII)
13. JIS C 6226 (1983) Code of the Japanese graphic character set for information interchange Japan institute for standardization
14. NIST SP 800-73（2002）Interfaces for personal identity verification
15. TIA/EIA-232-F (1997,R2002) Interface between data terminal equipment and data communication equipment employing serial binary data interchange
16. TIA/EIA-485-A (1998,R2003) Standard for electrical characteristics of generators and receivers for use in balanced digital multipoint systems