JÖNKÖPINGS KOMMUN

Jönköpings Kommuns

PROJEKTERINGSANVISNINGAR

FÖR STYR OCH ÖVERVAKNINGSINSTALLATIONER

vid ny- eller ombyggnation i egen regi

Bilaga 1.3 Anvisning SCADA

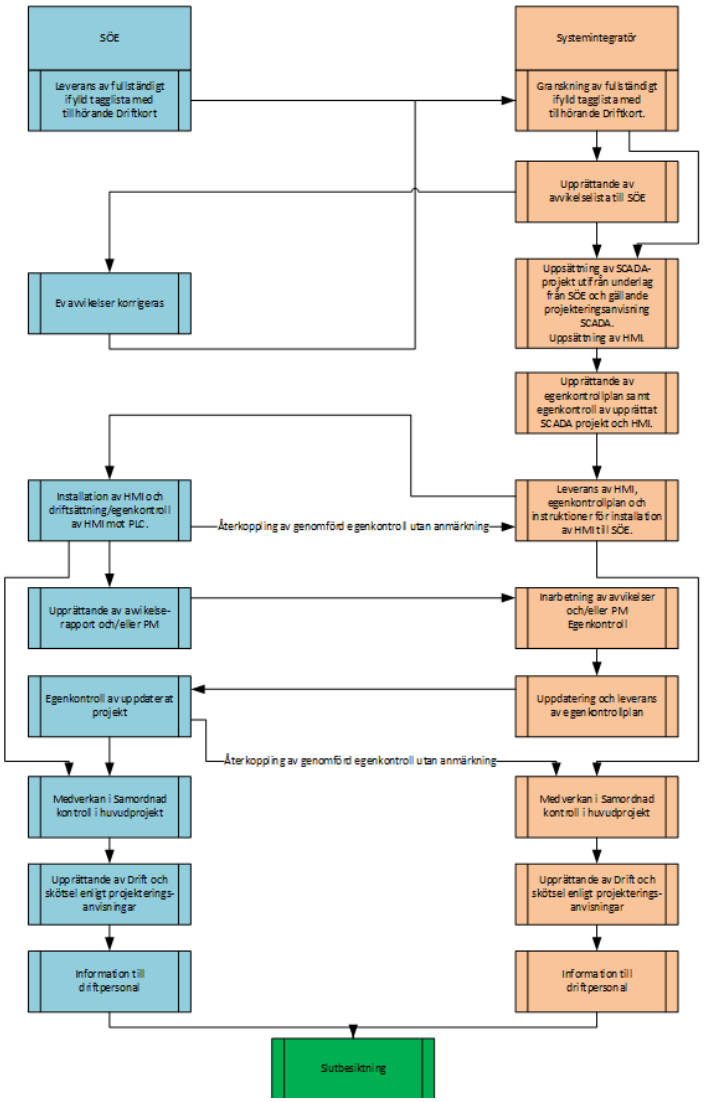
Upprättad 2020-04-24

INNEHÅLLSFÖRTECKNING

ALLMÄNT		Signering
8 Styr- och övervakningssystem	Denna handling är en bilaga till Jönköpings kommuns projekteringsanvisningar för Styr och övervakning bilaga 1. Denna beskrivning behandlar anslutning och integrering av det underordnade styr och övervakningssystemet på processnivå till det överordnade SCADA-systemet, Siemens Desigo CC, inom Jönköpings kommuns fastighetsbestånd.	
OMFATTNING	De lokala systemen, såsom PLC, skall kommunicera med SCADA, Desigo CC via protokollspecifika drivrutiner som tillhandahålls av styr och övervaknings entreprenören. För kommunikationen mellan noder för styr och övervakning nyttjas Jönköpings kommuns nätverk vilket är typ Ethernet. För kommunikation nyttjas protokoll ingående i TCP/IP. Jönköpings Kommun har valt BACnet som enda kommunikationsprotokoll för PLC mot överordnat SCADA-system DESIGO CC, eftersom det är en internationell standard för byggnadsautomation enligt ISO 16484-5 och för att protokollet är fabrikantsoberoende.	
BAKGRUND	B hanterar överordnat styr och övervakningssystem som ett gemensamt system som ska kunna nås från valfri plats inom B:s datanät. Detta ställer särskilda krav på ändringar och utökningar av systemet. Systemintegratören har tillsammans med B tagit fram denna integrationsstandard som styrande dokument till projektörer, entreprenörer och för upphandling av styr- och övervakningssystem. Övervakningssystemet är Siemens DESIGO CC vilket är placerat inom B:s infrastruktur. Systemet är installerat i B:s serverhall, alla styrsystem ska kopplas till DESIGO CC. Kommunikation mellan enheter ska ske med TCP/IP via B:s nätverk.	
SYFTE	Denna Integrationsstandard beskriver kraven för integration av styr- och övervakningsprojekt inom B:s fastighetsbestånd. Vid upphandling av styrsystem för fastighetsautomation skall standarden gälla som utökad anvisning vid upprättande av förfrågningsunderlag. Integrationsstandarden skall användas för att upprätthålla en jämn standard och kvalité i systemuppbyggnad och operatörsmiljö. Integrationsstandarden skall även användas som anvisning för projektörer och integratören vid ändring, utökning och nyinstallation av styr- och övervakningssystem inom B:s fastighetsbestånd. För en intelligent och enkel integration mot överordnat system DESIGO CC krävs det att leverantör av styr och övervakningssystem inom Jönköpings Kommun levererar produkter och system som uppfyller de minimi krav som ställs i BACnet standarden och därutöver några till. Funktionaliteten och uppbyggnaden på fältnivå varierar beroende på vilken PLC, eller vilket styrsystem som kopplas till systemet, detta dokument ska säkerställa att en lägsta nivå skapas med ett enhetligt handhavande i användargränssnittet i SCADA systemet DESIGO CC.	
MÅL	De levererade styr- och övervakningssystemens utformning skall inte skilja sig åt mellan olika entreprenörer. Detta ska åstadkommas genom att Styr och övervakningsentreprenören SÖE får tillgång till en utarbetad struktur samt ett fastlagt regelverk att följa. Med hjälp av detta kan entreprenörer leverera PLC som är strukturerade och programmerade för minimalt integrations och anpassningsarbete till DESIGO CC Alla styrsystem inom B:s fastighetsbestånd ska använda det gemensamma övervakningssystemet DESIGO CC.	

RUTIN FÖR INFORMATIONSUTBYTE MELLAN SÖE OCH SI

Nedanstående flödesschema beskriver hur systemintegrationsprojektets olika delmoment skall hanteras gällande samordningsansvarig och utförandeansvarig för Styr och övervakningsentreprenör (SÖE) och systemintegratör (SI)



BESTÄLLAREN

Beställaren i samråd med systemintegratören avgör likvärdighet vid val av utrustning och beslutar om eventuella avsteg från denna handling.

SYSTEMINTEGRATÖR SI

Systemintegratörens ansvar:

- Upprätta och vidareutveckla denna integrationsstandard
- Drift & underhåll av överordnade system DESIGO CC.
- Kontroll och integration av nya projekt enligt projektets tidplan.
- Tillhandahålla nödvändig information om systemet till entreprenören.
- Tillhandahålla IP adresser, nätmask, etc.
- Kontakter med B:s IT avdelning gällande servrar och IT relaterade frågor.

PROJEKTÖREN

Som projektör för styr- och övervaknings anläggningar och elanläggningar, fungerar denna standard som grundkrav vid upprättande av förfrågningsunderlag till upphandling av system som ska anslutas mot DESIGO CC.

	Projektörer skall: • Anpassa projektering mot denna standard • Projektanpassa kommunikationsgränssnittet • Ange anslutningspunkter för styrsystemen • Ange de mätpunkter som skall trendloggas	
ENTREPRENÖREN SÖE	Entreprenören levererar och installerar styr- och övervakningsanläggningar i enighet med denna integrationsstandard och gällande projekteringsanvisningar. **Entreprenören skall:** • Leverera och installera styr- och övervakningsanläggningar så att krav i denna integrationsstandard efterföljs. • Inhämta gällande nödvändiga uppgifter från systemintegratören	
FABRIKATVAL	Fabrikatsval B ställer krav på av PLC i syfte att inte hamna i en framtida beroendeställning till ett specifikt fabrikat, leverantör eller entreprenör. **B:s krav på val av fabrikat:** PLC ska kommunicera på BACnet IP och vara : • BTL certifierade enligt EN-ISO 16484-5 • Fabrikanten har en egen säljorganisation eller återförsäljare i Sverige. • Fabrikanten/återförsäljaren har supportorganisation i Sverige. • Fabrikanten/återförsäljaren har ett utbildningsprogram för produkten/systemet.	
INTEGRATION		
KRAV KOMMUNIKATION BACnet	B har valt BACnet/IP eftersom detta är ett standardiserat protokoll inom fastighetsautomation och som inte är fabrikantsberoende. BACnet är en fristående organisation som utfärdar och certifierar de produkter som använder BACnet för att säkerställa en hög sannolikhet för interoperabillitet mellan enheterna. BACnet är en internationell standard för byggnadsautomation enligt • ISO 16484-5 internationell • EN/ISO 16484-5 europeisk • ANSI/ASHRAE 135-2004 amerikansk • För mer information angående BACnet se följande sidor: • BACnet Intrest Group Europé (http://www.big-eu.org) • BACnet Internantional (http://www.bacnetinternational.org) • BACnet Official website (http://www.bacnet.org) BACnet är ett objektorienterat ”Client-Server” protokoll, det vill säga klienten frågar efter data eller utförande av en tjänst. Varje enhet kan vara en klient eller server. BACnet är ett ”event” och objektorienterat system det vill säga ingen kommunikation pågår i bakgrunden utan endast när det händer något i systemet som t.ex. larm. Det finns standardiserade objekt för bl.a. larm, trend och tidhantering. Vilket innebär att det alltid finns en gemensam databas så att informationen alltid är den samma oavsett var man är i systemet.	
KRAV FÖR SYSTEMINTEGRATIONEN	Styr och övervakningsentreprenören SÖE skall leverera en EDE-fil som underlag för import till systemintegratören SI samt PICS (Protocol Implementation Conformance Statement) och BIBB’s (BACnet Interoperability Building Blocks) dokument över sin produkt. I dessa listor skall det klart framgå: • Varje enhets BACnet adress (Device Object Instance). • BACnet adresslista över de värden som finns i anläggningen/aggregatet. • För varje BACnet object redovisas Object Type, Object Instance och beskrivning.	

Exempel på BACnet - EDE fil i csv format.
Exempel på struktur i EDE-fil.

# Example EDE-file.															
#															
#															
PROJECT_NAME	POL908														
VERSION_OF_REFERENCEFILE	1														
TIMESTAMP_OF_LAST_CHANGE															
AUTHOR_OF_LAST_CHANGE	Siemens Kristianstad														
VERSION_OF_LAYOUT	2														
keyname	device obj -instance	object-name	object-type	object-instance	description	supports COV	unit-code	hi-limit	low-limit	state-text	min-present-value	max-pres	unit-text	user-designation	
#															
# Parent Hierarchy levels															
#															
B1		B1			Byggnad 01										
B1A		B1A			Ventilation										
#															
# Common values															
#															
B1A'Ahu03		B1A'Ahu03			5507-57-01 Ventilation										
B1A'Ahu03'POL908	1001	B1A'Ahu03'POL908	8	1001	Climatic POL908	N									
B1A'Ahu03'AlmAck	1001	B1A'Ahu03'AlmAck	19	39100	5507-57-01Kvittera/Återställ larm	Y				20					
B1A'Ahu03'TSu	1001	B1A'Ahu03'TSu	0	40550	5507-57-01-GT1 Tilluftstemperatur	Y	62	99.000000	-64.000000		-64.000000		99.0000	C	
B1A'Ahu03'TEx	1001	B1A'Ahu03'TEx	0	28256	5507-57-01-GT3.2 Från/uftstemperatur	Y	62	99.000000	-64.000000		-64.000000		99.0000	C	
B1A'Ahu03'TOa	1001	B1A'Ahu03'TOa	0	53218	5507-57-01-GT3 Avluftstemperatur	Y	62	99.000000	-64.000000		-64.000000		99.0000	C	
B1A'Ahu03'TEh	1001	B1A'Ahu03'TEh	0	40895	5507-57-01-GT4 Avluftstemperatur	Y	62	99.000000	-64.000000		-64.000000		99.0000	C	
B1A'Ahu03'TFc	1001	B1A'Ahu03'TFc	0	44738	5507-57-01-GT5 Tilluftstemperatur efter VVX	Y	62	99.000000	-64.000000		-64.000000		99.0000	C	
B1A'Ahu03'TOaCom	1001	B1A'Ahu03'TOaCom	2	1881	5507-57-01-Utetemperatur - Kommunikation	Y	62							C	
B1A'Ahu03'FiDa	1001	B1A'Ahu03'FiDa	0	40720	5507-57-01-GP6:1 Filter uteluft	Y	53	142.136749	-40.000000		-100.000000		5000.00	Pa	
B1A'Ahu03'FiEx	1001	B1A'Ahu03'FiEx	0	13692	5507-57-01-GP6:2 Filter frånluft	Y	53	132.589636	-40.000000		-100.000000		5000.00	Pa	
B1A'Ahu03'SFP	1001	B1A'Ahu03'SFP	2	12177	5507-57-01 SFP	Y	95	100000.00000	-5.000000						
B1A'Ahu03'DmpShotOa	1001	B1A'Ahu03'DmpShotOa	4	6170	5507-57-01-ST2.1 Manöver utelufts pjäll	Y				9					
B1A'Ahu03'DmpShotEh	1001	B1A'Ahu03'DmpShotEh	4	43251	5507-57-01-ST2.2 Manöver avluftspjäll	Y				9					
B1A'Ahu03'Mnn3Svi	1001	B1A'Ahu03'Mnn3Svi	19	31604	5507-57-01 Serviceomkopplare	Y				14					
B1A'Ahu03'OpPosMan	1001	B1A'Ahu03'OpPosMan	13	9442	Drifomkopplare (EMS)	Y				13					
B1A'Ahu03'OpSta	1001	B1A'Ahu03'OpSta	19	32321	Aktuell driftstatus	Y				15					
B1A'Ahu03'OpMod	1001	B1A'Ahu03'OpMod	19	6080	Aktuellt driftläge	Y				16					
B1A'Ahu03'FanOpMod	1001	B1A'Ahu03'FanOpMod	19	28279	Aktuellt fläktsteg	Y				11					
B1A'Ahu03'CalendarEx	1001	B1A'Ahu03'CalendarEx	6	38174	Kalender undantag	N									
B1A'Ahu03'CalendarDit	1001	B1A'Ahu03'CalendarDit	6	51936	Kalender från	N									
B1A'Ahu03'Sched	1001	B1A'Ahu03'Sched	17	31059	Tidprogram	Y				12					
#															
# Supply air fan - common data points															
#															
B1A'Ahu03'FanSu		B1A'Ahu03'FanSu			Tilluftfläkt										
B1A'Ahu03'FanSu'PI	1001	B1A'Ahu03'FanSu'PI	0	9457	5507-57-01-TF01 Flöde	Y	1000	40000.00000	-100.000000		-100.00000	150000.0	l/s		
B1A'Ahu03'FanSu'PMdl	1001	B1A'Ahu03'FanSu'PMdl	1	59037	5507-57-01-TF01 Modulerande	Y	98	100.000000	0.000000		0.000000	100.0000	%		

För mera information och beskrivning över EDE filen och dess innehåll se BACnet Intrest Group Europe (http://www.big-eu.org/service/software.php

KOMMUNIKATIONS-PRINCIPER

Informationsnivå *(DESIGO CC)*
Med informationsnivå avses kommunikation mellan server och klienter i det överordnat styr- och övervakningssystemet.

Processnivå *(PLC)*
Med processnivå avses kommunikation mellan fältplacerade PLC och styr- och övervakningsenheter.
Kommunikation mellan PLC och server skall ske via BACnet/IP över nätverk.

Fältnivå *(Lokala styrenheter, typ rumsreglering etc.)*
Med fältnivå avses kommunikation mellan fältplacerade styr- och övervakningskomponenter och mindre styrenheter så som rumsregulatorer eller frekvensomformare. Kommunikation mellan processnivå och lokalt placerade enheter kan utföras med BACnet, Modbus, M-bus eller likvärdiga protokoll.

Kommunikation via Ethernet
All kommunikation mellan utrustning skall ske via B:s befintliga IT-infrastruktur. Inga separata fabrikatsspecifika kommunikationsservrar (som är att betrakta som systemspecifikt SCADA system) får förekomma.

B tillhandahåller RJ45 uttag placerade i korskopplingsställ alternativt framdraget till apparatrum. Allt Ethernet kablage från anslutningspunkten och ut i anläggningen ingår i respektive projekt.

GRUNDLÄGGANDE KRAV PÅ BACnet-OBJEKT

För certifiering av en BACnet produkt krävs det av fabrikant uppfyllande av standarden på ett antal punkter som är ”requierd” i standarden.
Av SÖE Entreprenör levererade PLC:ar i detta projekt ska förutom kraven för BTL certifering även minimum uppfylla kraven i, Ashrae Standard BACnet Object.
Det skall t.ex. vara möjligt för användaren att från DESIGO CC kunna ändra larmprioritet på ett larmobjekt genom att byta/ändra inställt BACnet Notification Class Object.

BYGGNADS/PROGRAM-STRUKTUR I PLC

Obect name (objektnamn) ska byggas upp så att en trädstruktur erhålles i DESIGO CC.
Description (klartextbeskrivning) av signalens funktion, presenteras med fullständig systembeteckning för respektive signal för presentation i flödesbild i överordnat system.
Exempel B1'A'Ahu03'Hcl'TFrPrt

Område	*5507*
B1 = Byggnad	*01*
A= Ventilation(System)	*57*
Ahu03 = Aggregat(Undersystem)	*04*
Hcl = Luftvärmare(Objekt)	*Luftvärmare*
TFrPrt = Frysskydd(Komponent)	

	~~5507~~-57-01-04-GT8 Se även exempel på EDE fil under rubrik ”KRAV FÖR SYSTEMINTEGRATIONEN”	
KOMMUNIKATION	Krav på PLC är att de ska hantera BACnet/IP, Device Object ID, Device name, Network ID, IP-adress, IP-port samt BBMD.	
LARM/LARMKVITTERING	Larm ska kunna hanteras och kvitteras från DESIGO CC vilket innebär att kraven under rubriken,”ASHRAE STANDARD BACNET OBJECT” ska stödjas i levererad PLC. Larm definieras som: • Intrinsic Reporting, (rekomenderas) innebär att larmegenskaper är definierade på respektive BACnet- objekt. • eller Event Enrollment Object, dvs. separata BACnet-larmobjekt Samtliga larmpunkter skall i BACnet enhet vara programmerade med klartextinformation: Med klartext information menas ex. ”5507-57-01-04-GT8 Utlöst frysskydd vatten”	
TREND/LOGGNING	Trender och Loggningar ska kunna hanteras från DESIGO CC, vilket innebär att lokala trendobjekt måste stödjas i levererad PLC.	
HANDSTÄLLNING AV OBJEKT	Värdesändring från DESIGO CC ska ske på BACnet kommandoprioritet 16. Handställning ska ske på BACnet prioritet 8.	
TIDKATALOGER	BACnet standardens tidhantering gäller med lokala tidkataloger i PLC.	
MÄTINSAMLING	Energimätning ska ske via PLC, detta gäller även loggning av andra värden som kan vara intressanta ur driftsynpunkt. Vilket innebär att lokala trendobjekt måste stödjas i levererad PLC. (Offline trendobjekt)	
LÄS OCH SKRIVRÄTTIGHETER	Läs och skrivrättigheter i PLC ska vara enligt rubriken nedan: ”ASHRAE STANDARD BACNET OBJECT”	
UTDRAG UR ”ASHRAE STANDARDEN BACNET OBJECT ”		
ASHRAE STANDARD FÖR BACNET OBJECT	I nedan angivna BACnet object är minikravet för en BTL certifiering av en produkt att den minst uppfyller requierd (R) för ett godkännande. Dock är det så att produkter som ligger på minimnivå vid en certifering, kan innebära att funktionaliteten i det överordnade systemet kan begränsas. För att säkertställa en smidig integration och hög funktionalitet krävs det att respektive SÖE entreprenör nyttjar de möjligheterna BACnet standarden erbjuder och följer dessa anvisningar. För eventuella avsteg från kraven ska godkännande inhämtas av B och SI. Nedan listade BACnet Object är tagna direkt ur BACnet Standarden, under varje object framgår grundkraven för BTL Certifiering av en produkt med ett R(requierd) i kolumnen till höger. Utöver det så är det som är markerat med grön överstrykning också ett krav att uppfylla. ***Accumulator Object Type*** The Accumulator object type defines a standardized object whose properties represent the externally visible characteristics of a device that indicates measurements made by counting pulses. This object maintains precise measurement of input count values, accumulated over time. The accumulation of pulses represents the measured quantity in unsigned integer units. This object is also concerned with the accurate representation of values presented on meter read-outs. This includes the ability to initially set the Present_Value property to the value currently displayed by the meter (as when the meter is installed), and to duplicate the means by which it is advanced, including simulating a modulo-N divider prescaling the actual meter display value, as shown in Figure 12-1.	

Typical applications of such devices are in peak load management and in accounting and billing management systems. This object is not intended to meet all such applications. Its purpose is to provide information about the quantity being measured, such as electric power, water, or natural gas usage, according to criteria specific to the application.

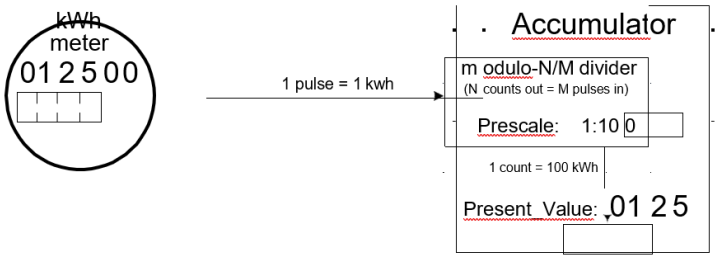


Figure 12-1. Example of an Accumulator object

Table 12-1. Properties of the Accumulator Object

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Present_Value	Unsigned	R[1]
Description	CharacterString	O
Device_Type	CharacterString	O
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	R
Reliability	BACnetReliability	O
Out_Of_Service	BOOLEAN	R
Scale	BACnetScale	R
Units	BACnetEngineeringUnits	R
Prescale	BACnetPrescale	O
Max_Pres_Value	Unsigned	R
Value_Change_Time	BACnetDateTime	O[2]
Value_Before_Change	Unsigned	O[2,3]
Value_Set	Unsigned	O[2,3]
Logging_Record	BACnetAccumulatorRecord	O
Logging_Object	BACnetObjectIdentifier	O
Pulse_Rate	Unsigned	O[1,4]
High_Limit	Unsigned	O[4]
Low_Limit	Unsigned	O[4]
Limit_Monitoring_Interval	Unsigned	O[4]
Notification_Class	Unsigned	O[4]
Time_Delay	Unsigned	O[4]
Limit_Enable	BACnetLimitEnable	O[4]
Event_Enable	BACnetEventTransitionBits	O[4]
Acked_Transitions	BACnetEventTransitionBits	O[4]
Notify_Type	BACnetNotifyType	O[4]
Event_Time_Stamps	BACnetARRAY[3] of	O[4]
Profile_Name	CharacterString	O

1 This property is required to be writable when Out_Of_Service is TRUE.
2 These properties are required if either Value_Before_Change or Value_Set is writable.
3 Either Value_Before_Change or Value_Set may be writable, but not both.
4 These properties are required if the object supports intrinsic reporting.

Analog Input Object Type

The Analog Input object type defines a standardized object whose properties represent the externally visible characteristics of an analog input. The object and its properties are
summarized in Table 12-2 and described in detail in this subclause.

Table 12-2. Properties of the Analog Input Object Type

Property Identifier	Property Datatype	Conformance
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Present_Value	REAL	R[1]
Description	CharacterString	O
Device_Type	CharacterString	O
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	R
Reliability	BACnetReliability	O
Out_Of_Service	BOOLEAN	R
Update_Interval	Unsigned	O
Units	BACnetEngineeringUnits	R
Min_Pres_Value	REAL	O
Max_Pres_Value	REAL	O
Resolution	REAL	O
COV_Increment	REAL	O[2]
Time_Delay	Unsigned	O[3]
Notification_Class	Unsigned	O[3]
High_Limit	REAL	O[3]
Low_Limit	REAL	O[3]
Deadband	REAL	O[3]
Limit_Enable	BACnetLimitEnable	O[3]
Event_Enable	BACnetEventTransitionBits	O[3]
Acked_Transition	BACnetEventTransitionBits	O[3]
Notify_Type	BACnetNotifyType	O[3]
Event_Time_Stam	BACnetARRAY[3] of	O[3]
Profile_Name	CharacterString	O

1 This property is required to be writable when Out_Of_Service is TRUE.
2 This property is required if the object supports COV reporting.
3 These properties are required if the object supports intrinsic reporting.

Analog Output Object Type
The Analog Output object type defines a standardized object whose properties represent the externally visible characteristics of an analog output. The object and its properties are
summarized in Table 12-3 and described in detail in this subclause.

Table 12-3. Properties of the Analog Output Object Type

Property	Property Datatype	Conformance
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Present_Value	REAL	W
Description	CharacterString	O
Device_Type	CharacterString	O
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	R
Reliability	BACnetReliability	O
Out_Of_Service	BOOLEAN	R
Units	BACnetEngineeringUnits	R
Min_Pres_Value	REAL	O
Max_Pres_Value	REAL	O
Resolution	REAL	O
Priority_Array	BACnetPriorityArray	R
Relinquish_Defa	REAL	R
COV_Increment	REAL	O[1]
Time_Delay	Unsigned	O[2]
Notification_Cla	Unsigned	O[2]
High_Limit	REAL	O[2]
Low_Limit	REAL	O[2]
Deadband	REAL	O[2]
Limit_Enable	BACnetLimitEnable	O[2]
Event_Enable	BACnetEventTransitionBits	O[2]
Acked_Transitio	BACnetEventTransitionBits	O[2]
Notify_Type	BACnetNotifyType	O[2]
Event_Time_Sta	BACnetARRAY[3] of	O[2]
Profile_Name	CharacterString	O

1 This property is required if the object supports COV reporting.
2 These properties are required if the object supports intrinsic reporting.

Analog Value Object Type
The Analog Value object type defines a standardized object whose properties represent the externally visible characteristics of an analog value. An "analog value" is a control system parameter residing in the memory of the BACnet Device. The object and its properties are summarized in Table 12-4 and described in detail in this subclause.

Table 12-4. Properties of the Analog Value Object Type

Property Identifier	Property Datatype	Conformance
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Present_Value	REAL	R[4]
Description	CharacterString	O
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	R
Reliability	BACnetReliability	O
Out_Of_Service	BOOLEAN	R
Units	BACnetEngineeringUnit	R
Priority_Array	BACnetPriorityArray	O[1]
Relinquish_Default	REAL	O[1]
COV_Increment	REAL	O[2]
Time_Delay	Unsigned	O[3]
Notification_Class	Unsigned	O[3]
High_Limit	REAL	O[3]
Low_Limit	REAL	O[3]
Deadband	REAL	O[3]
Limit_Enable	BACnetLimitEnable	O[3]
Event_Enable	BACnetEventTransition	O[3]
Acked_Transitions	BACnetEventTransition	O[3]
Notify_Type	BACnetNotifyType	O[3]
Event_Time_Stamps	BACnetARRAY[3] of	O[3]
Profile_Name	CharacterString	O

1 If Present_Value is commandable, then both of these properties shall be present.
2 This property is required if the object supports COV reporting.
3 These properties are required if the object supports intrinsic reporting.
4 If Present_Value is commandable, then it is required to be writable. This property is required to be writable when Out_Of_Service is TRUE.

Averaging Object Type
The Averaging object type defines a standardized object whose properties represent the externally visible characteristics of a value that is sampled periodically over a specified time interval. The Averaging object records the minimum, maximum and average value over the interval, and makes these values visible as properties of the Averaging object. The sampled value may be the value of any BOOLEAN, INTEGER, Unsigned, Enumerated or REAL property value of any object within the BACnet Device in which the object resides. Optionally, the object property to be sampled may exist in a different BACnet Device. The Averaging object shall use a "sliding window" technique that maintains a buffer of *N* samples distributed over the specified interval. Every (time interval/N) seconds a new sample is recorded displacing the oldest sample from the buffer. At this time, the minimum, maximum and average are recalculated. The buffer shall maintain an indication for each sample that permits the average calculation and minimum/maximum algorithm to determine the number of valid samples in the buffer. The Averaging object type and its properties are summarized in Table 12-5 and described in detail in this subclause.

Table 12-5. Properties of the Averaging Object Type

Property	Property	Conformance
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Minimum_Value	REAL	R
Minimum_Value_Ti	BACnetDateTime	O
Average_Value	REAL	R
Variance_Value	REAL	O
Maximum_Value	REAL	R
Maximum_Value_Ti	BACnetDateTime	O
Description	CharacterString	O
Attempted_Samples	Unsigned	W[1]
Valid_Samples	Unsigned	R
Object_Property_Ref	BACnetDeviceObjectPropert	R[1]
Window_Interval	Unsigned	W[1]
Window_Samples	Unsigned	W[1]
Profile_Name	CharacterString	O

1 If any of these properties are written to using BACnet services, then all of the buffer samples shall become invalid, 'Attempted_Samples' shall become zero, 'Valid_Samples' shall become zero, 'Minimum_Value' shall become INF, 'Average_Value' shall become NaN and 'Maximum_Value' shall become -INF.

Binary Input Object Type

The Binary Input object type defines a standardized object whose properties represent the externally visible characteristics of a binary input. A "binary input" is a physical device or hardware input that can be in only one of two distinct states. In this description, those states are referred to as ACTIVE and INACTIVE. A typical use of a binary input is to indicate whether a particular piece of mechanical equipment, such as a fan or pump, is running or idle. The state ACTIVE corresponds to the situation when the equipment is on or running, and INACTIVE corresponds to the situation when the equipment is off or idle. In some applications, electronic circuits may reverse the relationship between the application-level logical states ACTIVE and INACTIVE and the physical state of the underlying hardware. For example, a normally open relay contact may result in an ACTIVE state when the relay is energized, while a normally closed relay contact may result in an INACTIVE state when the relay is energized. The Binary Input object provides for this possibility by including a Polarity property. See 12.6.4 and 12.6.11.
The object and its properties are summarized in Table 12-6 and described in detail in this subclause.

Table 12-6. Properties of the Binary Input Object Type

Property Identifier	Property Datatype	Conformance
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Present_Value	BACnetBinaryPV	R[1]
Description	CharacterString	O
Device_Type	CharacterString	O
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	R
Reliability	BACnetReliability	O
Out_Of_Service	BOOLEAN	R
Polarity	BACnetPolarity	R
Inactive_Text	CharacterString	O[2]
Active_Text	CharacterString	O[2]
Change_Of_State_Ti	BACnetDateTime	O[3]
Change_Of_State_Co	Unsigned	O[3]
Time_Of_State_Count	BACnetDateTime	O[3]
Elapsed_Active_Time	Unsigned32	O[4]
Time_Of_Active_Tim	BACnetDateTime	O[4]
Time_Delay	Unsigned	O[5]
Notification_Class	Unsigned	O[5]
Alarm_Value	BACnetBinaryPV	O[5]
Event_Enable	BACnetEventTransition	O[5]
Acked_Transitions	BACnetEventTransition	O[5]
Notify_Type	BACnetNotifyType	O[5]
Event_Time_Stamps	BACnetARRAY[3] of	O[5]
Profile_Name	CharacterString	O

1 This property is required to be writable when Out_Of_Service is TRUE.
2 If one of the optional properties Inactive_Text or Active_Text is present, then both of these properties shall be present.
3 If one of the optional properties Change_Of_State_Time, Change_Of_State_Count, or Time_Of_State_Count_Reset is present, then all of these properties shall be present.
4 If one of the optional properties Elapsed_Active_Time or Time_Of_Active_Time_Reset is present, then both of these properties shall be present.
5 These properties are required if the object supports intrinsic reporting.

Binary Output Object Type
The Binary Output object type defines a standardized object whose properties represent the externally visible characteristics of a binary output. A "binary output" is a physical device or hardware output that can be in only one of two distinct states. In this description, those states are referred to as ACTIVE and INACTIVE. A typical use of a binary output is to switch a particular piece of mechanical equipment, such as a fan or pump, on or off. The state ACTIVE corresponds to the situation when the equipment is on or running, and INACTIVE corresponds to the situation when the equipment is off or idle.
In some applications, electronic circuits may reverse the relationship between the application-level logical states, ACTIVE and INACTIVE, and the physical state of the underlying hardware. For example, a normally open relay contact may result in an ACTIVE state (device energized) when the relay is energized, while a normally closed relay contact may result in an ACTIVE state (device energized) when the relay is not energized. The Binary Output object provides for this possibility by including a Polarity property. See 12.7.4 and 12.7.11.
The object and its properties are summarized in Table 12-8 and described in detail in this subclause.

Table 12-8. Properties of the Binary Output Object Type

Property Identifier	Property Datatype	Conformance
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Present_Value	BACnetBinaryPV	W
Description	CharacterString	O
Device_Type	CharacterString	O
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	R
Reliability	BACnetReliability	O
Out_Of_Service	BOOLEAN	R
Polarity	BACnetPolarity	R
Inactive_Text	CharacterString	O[1]
Active_Text	CharacterString	O[1]
Change_Of_State_Ti	BACnetDateTime	O[2]
Change_Of_State_Co	Unsigned	O[2]
Time_Of_State_Count	BACnetDateTime	O[2]
Elapsed_Active_Time	Unsigned32	O[3]
Time_Of_Active_Tim	BACnetDateTime	O[3]
Minimum_Off_Time	Unsigned32	O
Minimum_On_Time	Unsigned32	O
Priority_Array	BACnetPriorityArray	R
Relinquish_Default	BACnetBinaryPV	R
Time_Delay	Unsigned	O[4]
Notification_Class	Unsigned	O[4]
Feedback_Value	BACnetBinaryPV	O[4]
Event_Enable	BACnetEventTransitionBit	O[4]
Acked_Transitions	BACnetEventTransitionBit	O[4]
Notify_Type	BACnetNotifyType	O[4]
Event_Time_Stamps	BACnetARRAY[3] of	O[4]
Profile_Name	CharacterString	O

1 If one of the optional properties Inactive_Text or Active_Text is present, then both of these properties shall be present.
2 If one of the optional properties Change_Of_State_Time, Change_Of_State_Count, or Time_Of_State_Count_Reset is present, then all of these properties shall be present.
3 If one of the optional properties Elapsed_Active_Time or Time_Of_Active_Time_Reset is present, then both of these properties shall be present.
4 These properties are required if the object supports intrinsic reporting.

Binary Value Object Type

The Binary Value object type defines a standardized object whose properties represent the externally visible characteristics of a binary value. A "binary value" is a control system parameter residing in the memory of the BACnet Device. This parameter may assume only one of two distinct states. In this description, those states are referred to as ACTIVE and INACTIVE. The Binary Value object and its properties are summarized in Table 12-10 and described in detail in this subclause.

Table 12-10. Properties of the Binary Value Object Type

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Present_Value	BACnetBinaryPV	R[1]
Description	CharacterString	O
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	R
Reliability	BACnetReliability	O
Out_Of_Service	BOOLEAN	R
Inactive_Text	CharacterString	O[2]
Active_Text	CharacterString	O[2]
Change_Of_State_Time	BACnetDateTime	O[3]
Change_Of_State_Count	Unsigned32	O[3]
Time_Of_State_Count_Res	BACnetDateTime	O[3]
Elapsed_Active_Time	Unsigned32	O[4]
Time_Of_Active_Time_Re	BACnetDateTime	O[4]
Minimum_Off_Time	Unsigned32	O
Minimum_On_Time	Unsigned32	O
Priority_Array	BACnetPriorityArray	O[5]
Relinquish_Default	BACnetBinaryPV	O[5]
Time_Delay	Unsigned	O[6]
Notification_Class	Unsigned	O[6]
Alarm_Value	BACnetBinaryPV	O[6]
Event_Enable	BACnetEventTransition	O[6]
Acked_Transitions	BACnetEventTransition	O[6]
Notify_Type	BACnetNotifyType	O[6]
Event_Time_Stamps	BACnetARRAY[3] of	O[6]
Profile_Name	CharacterString	O

1 If Present_Value is commandable, then it is required to be writable. This property is required to be writable when
Out_Of_Service is TRUE.
2 If one of the optional properties Inactive_Text or Active_Text is present, then both of these properties shall be present.
3 If one of the optional properties Change_Of_State_Time, Change_Of_State_Count, or Time_Of_State_Count_Reset is present, then all of these properties
shall be present.
4 If one of the optional properties Elapsed_Active_Time or Time_Of_Active_Time_Reset is present, then both of these properties shall be present.

5 If Present_Value is commandable, then both of these properties shall be present.
6 These properties are required if the object supports intrinsic reporting.

Calendar ObjectType
The Calendar object type defines a standardized object used to describe a list of calendar dates, which might be thought of as "holidays," "special events," or simply as a list of dates. The object and its properties are summarized in Table 12-11 and described in detail in this subclause.

Table 12-11. Properties of the Calendar Object Type

Property Identifier	Property Datatype	Conformance
Object_Identifier	BACnetObjectIdentifie	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Description	CharacterString	O
Present_Value	BOOLEAN	R
Date_List	List of	R
Profile_Name	CharacterString	O

Command Object Type

Table 12-12. Properties of the Command Object Type

Property	Property Datatype	Conformance
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Description	CharacterString	O
Present_Value	Unsigned	W
In_Process	BOOLEAN	R
All_Writes_Succ	BOOLEAN	R
Action	BACnetARRAY[N] of	R
Action_Text	BACnetARRAY[N] of	O
Profile_Name	CharacterString	O

Device Object Type

The Device object type defines a standardized object whose properties represent the externally visible characteristics of a BACnet Device. There shall be exactly one Device object in each BACnet Device. A Device object is referenced by its Object_Identifier property, which is not only unique to the BACnet Device that maintains this object but is also unique throughout the BACnet internetwork. The Device object type and its properties are summarized in Table 12-13 and described in detail in this subclause.

Table 12-13. Properties of the Device Object Type

Property Identifier	Property Datatype	Conformance
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
System_Status	BACnetDeviceStatus	R
Vendor_Name	CharacterString	R
Vendor_Identifier	Unsigned16	R
Model_Name	CharacterString	R
Firmware_Revision	CharacterString	R
Application_Softwar	CharacterString	R
Location	CharacterString	O
Description	CharacterString	O
Protocol_Version	Unsigned	R
Protocol_Revision	Unsigned	R
Protocol_Services_S	BACnetServicesSupported	R
Protocol_Object_T	BACnetObjectTypesSupporte	R
Object_List	BACnetARRAY[N]of	R
Max_APDU_Length	Unsigned	R
Segmentation_Suppo	BACnetSegmentation	R
Max_Segments_Acc	Unsigned	O[1]
VT_Classes_Support	List of BACnetVTClass	O[2]
Active_VT_Sessions	List of BACnetVTSession	O[2]
Local_Time	Time	O[3,4]
Local_Date	Date	O[3,4]
UTC_Offset	INTEGER	O[4]
Daylight_Savings_St	BOOLEAN	O[4]
APDU_Segment_Ti	Unsigned	O[1]
APDU_Timeout	Unsigned	R
Number_Of_APDU_	Unsigned	R
List_Of_Session_Ke	List of BACnetSessionKey	O
Time_Synchroniz	List of BACnetRecipient	O[5]
Max_Master	Unsigned(1..127)	O[6]
Max_Info_Frames	Unsigned	O[6]
Device_Address_Bin	List of	R
Database_Revision	Unsigned	R
Configuration_Files	BACnetARRAY[N] of	O[7]
Last_Restore_Time	BACnetTimeStamp	O[7]
Backup_Failure_Tim	Unsigned16	O[8]
Active_COV_Subscr	List of	O[9]
Slave_Proxy_Enable	BACnetArray[N] of	O[10]
Manual_Slave_Addr	List of	O[10]
Auto_Slave_Discove	BACnetArray[N] of	O[11]
Slave_Address_Bind	List of	O[12]
Profile_Name	CharacterString	O

1 Required if segmentation of any kind is supported.
2 If one of the properties VT_Classes_Supported or Active_VT_Sessions is present, then both of these properties shall be present. Both
properties are required if support for VT Services is indicated in the PICS.
3 If the device supports the execution of the TimeSynchronization service, then these properties shall be present.
4 If the device supports the execution of the UTCTimeSynchronization service, then these properties shall be present.
5 Required if PICS indicates that this device is a Time Master. If present, this property shall be writable.
6 These properties are required if the device is an MS/TP master node.
7 These properties are required if the device supports the backup and restore procedures.
8 This property must be present and writable if the device supports the backup and restore procedures.
9 This property is required if the device supports execution of either the SubscribeCOV or SubscribeCOVProperty service.
10 This property shall be present and writable if the device is capable of being a Slave- Proxy device.
11 This property shall be present if the device is capable of being a Slave-Proxy device that implements automatic discovery of slaves.
12 This property shall be present if the device is capable of being a Slave-Proxy device.

Event Enrollment Object Type
The Event Enrollment object type defines a standardized object that represents and contains the information required for managing events within BACnet systems. "Events" are changes of value of any property of any object that meet certain predetermined criteria. The primary purpose for Event Enrollment objects is to define an event and to provide a connection between the occurrence of an event and the transmission of a notification message to one or more recipients.
The Event Enrollment object contains the event-type description, the parameters needed to determine if the event has occurred, and a device to be notified. Alternatively, a Notification Class object may serve to identify the recipients of event notifications. A device is considered to be "enrolled for event notification" if it is the recipient to be notified or one of the recipients in a Notification Class object referenced by the Event Enrollment object.
Clause 13 describes the interaction between Event Enrollment objects and the Alarm and Event application services. The Event Enrollment object and its properties are summarized in Table 12-14 and described in detail in this subclause.

Table 12-14. Properties of the Event Enrollment Object Type

Property Identifier	Property Datatype	Conformanc
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Description	CharacterString	O
Event_Type	BACnetEventType	R
Notify_Type	BACnetNotifyType	R
Event_Parameters	BACnetEventParameter	R
Object_Property_Referen	BACnetDeviceObjectPro	R
Event_State	BACnetEventState	R
Event_Enable	BACnetEventTransition	R
Acked_Transitions	BACnetEventTransition	R
Notification_Class	Unsigned	R
Event_Time_Stamps	BACnetARRAY[3] of	R
Profile_Name	CharacterString	O

File Object Type
The File object type defines a standardized object that is used to describe properties of data files that may be accessed using File Services (see Clause 14). The file object type and its properties are summarized in Table 12-16 and described in detail in this subclause.

Table 12-16. Properties of the File Object Type

Property Identifier	Property Datatype	Conformance
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Description	CharacterString	O
File_Type	CharacterString	R
File_Size	Unsigned	R[1]
Modification_Date	BACnetDateTime	R
Archive	BOOLEAN	W
Read_Only	BOOLEAN	R
File_Access_Metho	BACnetFileAccessMeth	R
Record_Count	Unsigned	O[2]
Profile_Name	CharacterString	O

1 If the file size can be changed by writing to the file, and File_Access_Method is STREAM_ACCESS, then this property shall be writable.
2 This property shall be present only if File_Access_Method is RECORD_ACCESS. If the number of records can be changed by writing to the file, then this property shall be writable.

Group ObjectType
The Group object type defines a standardized object whose properties represent a collection of other objects and one or more of their properties. A group object is used to simplify the exchange of information between BACnet Devices by providing a shorthand way to specify all members of the group at once. A group may be formed using any combination of object types. The group object and its properties are summarized in Table 12-17 and described in detail in this subclause.

Table 12-17. Properties of the Group Object Type

Property Identifier	Property Datatype	Conformance
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Description	CharacterString	O
List_Of_Group_Me	List of	R
Present_Value	List of ReadAccessResult	R
Profile_Name	CharacterString	O

Life Safety Point Object Type
The Life Safety Point object type defines a standardized object whose properties represent the externally visible characteristics associated with initiating and indicating devices in fire, life safety and security applications. The condition of a Life Safety Point object is represented by a *mode* and a *state*.
Mode changes determine the object's inner logic and, consequently, influence the evaluation of the state. Typically, the operating *mode* would be under operator control.
The *state* of the object represents the condition of the controller according to the logic internal to the device. The implementation of the logic applied to such controllers to determine the various possible states is a local matter.
Typical applications of the Life Safety Point object include automatic fire detectors, pull stations, sirens, supervised printers, etc. Similar objects can be identified in security control panels.
The Life Safety Point object type and its properties are summarized in Table 12-18 and described in detail in this subclause.

Table 12-18. Properties of the Life Safety Point Object Type

Property Identifier	Property Datatype	Conformance
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Present_Value	BACnetLifeSafetyState	R[1]
Tracking_Value	BACnetLifeSafetyState	O
Description	CharacterString	O
Device_Type	CharacterString	O
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	R
Reliability	BACnetReliability	R[1]
Out_Of_Service	BOOLEAN	R
Mode	BACnetLifeSafetyMode	W
Accepted_Modes	List of BACnetLifeSafetyMode	R
Time_Delay	Unsigned	O[2]
Notification_Clas	Unsigned	O[2]
Life_Safety_Alar	List of BACnetLifeSafetyState	O[2]
Alarm_Values	List of BACnetLifeSafetyState	O[2]
Fault_Values	List of BACnetLifeSafetyState	O[2]
Event_Enable	BACnetEventTransitionBits	O[2]
Acked_Transition	BACnetEventTransitionBits	O[2]
Notify_Type	BACnetNotifyType	O[2]
Event_Time_Sta	BACnetARRAY [3] of	O[2]
Silenced	BACnetSilencedState	R
Operation_Expect	BACnetLifeSafetyOperation	R
Maintenance_Req	BACnetMaintenance	O
Setting	Unsigned8	O
Direct_Reading	REAL	O[3]
Units	BACnetEngineeringUnits	O[3]
Member_Of	List of	O
Profile_Name	CharacterString	O

1 These properties are required to be writable when Out_Of_Service is TRUE.
2 These properties are required if the object supports intrinsic alarming.
3 If either of these properties is present, then both must be present.

Life Safety Zone Object Type

The Life Safety Zone object type defines a standardized object whose properties represent the externally visible characteristics associated with an arbitrary group of BACnet Life Safety Point and Life Safety Zone objects in fire, life safety and security applications. The condition of a Life Safety Zone object is represented by a *mode* and a *state*.

Mode changes determine the object's inner logic and, consequently, influence the evaluation of the state. Typically, the operating *mode* would be under operator control.

The *state* of the object represents the condition of the controller according to the logic internal to the device. The implementation of the logic applied to such controllers to determine the various possible states is a local matter. Typical applications of the Life Safety Zone object include fire zones, panel zones, detector lines, extinguishing controllers, remote transmission controllers, etc. Similar objects can be identified in security control panels. The Life Safety Zone object type and its properties are summarized in Table 12-19 and described in detail in this subclause.

NOTE: Do not confuse the Present_Value state with the Event_State property, which reflects the offnormal state of the Life Safety Zone object.

Table 12-19. Properties of the Life Safety Zone Object Type

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Present_Value	BACnetLifeSafetyState	R[1]
Tracking_Value	BACnetLifeSafetyState	O
Description	CharacterString	O
Device_Type	CharacterString	O
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	R
Reliability	BACnetReliability	R[1]
Out_Of_Service	BOOLEAN	R
Mode	BACnetLifeSafetyMode	W
Accepted_Modes	List of BACnetLifeSafetyMode	R
Time_Delay	Unsigned	O[2]
Notification_Class	Unsigned	O[2]
Life_Safety_Alarm_	List of BACnetLifeSafetyState	O[2]
Alarm_Values	List of BACnetLifeSafetyState	O[2]
Fault_Values	List of BACnetLifeSafetyState	O[2]
Event_Enable	BACnetEventTransitionBits	O[2]
Acked_Transitions	BACnetEventTransitionBits	O[2]
Notify_Type	BACnetNotifyType	O[2]
Event_Time_Stamps	BACnetARRAY [3] of	O[2]
Silenced	BACnetSilencedState	R
Operation_Expected	BACnetLifeSafetyOperation	R
Maintenance_Requir	BOOLEAN	O
Zone_Members	List of	R
Member_Of	List of	O
Profile_Name	CharacterString	O

1 These properties are required to be writable when Out_Of_Service is TRUE.
2 These properties are required if the object supports intrinsic alarming.

Loop Object Type

The Loop object type defines a standardized object whose properties represent the externally visible characteristics of any form of feedback control loop. Flexibility is achieved by providing three independent gain constants with no assumed values for units. The appropriate gain units are determined by the details of the control algorithm, which is a local matter. The Loop object type and its properties are summarized in Table 12-20 and described in detail in this subclause. Figure 12-2 illustrates the relationship between the Loop object properties and the other objects referenced by the loop.

Table 12-20. Properties of the Loop Object Type

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Present_Value	REAL	R
Description	CharacterString	O
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	R
Reliability	BACnetReliability	O
Out_Of_Service	BOOLEAN	R
Update_Interval	Unsigned	O
Output_Units	BACnetEngineeringUnits	R
Manipulated_Variable_	BACnetObjectPropertyRefe	R
Controlled_Variable_R	BACnetObjectPropertyRefe	R
Controlled_Variable_V	REAL	R
Controlled_Variable_U	BACnetEngineeringUnits	R
Setpoint_Reference	BACnetSetpointReference	R
Setpoint	REAL	R
Action	BACnetAction	R
Proportional_Constant	REAL	O[1]
Proportional_Constant_	BACnetEngineeringUnits	O[1]
Integral_Constant	REAL	O[2]
Integral_Constant_Unit	BACnetEngineeringUnits	O[2]
Derivative_Constant	REAL	O[3]
Derivative_Constant_U	BACnetEngineeringUnits	O[3]
Bias	REAL	O
Maximum_Output	REAL	O
Minimum_Output	REAL	O
Priority_For_Writing	Unsigned	R
COV_Increment	REAL	O[4]
Time_Delay	Unsigned	O[5]
Notification_Class	Unsigned	O[5]
Error_Limit	REAL	O[5]
Event_Enable	BACnetEventTransitionBit	O[5]
Acked_Transitions	BACnetEventTransitionBit	O[5]
Notify_Type	BACnetNotifyType	O[5]
Event_Time_Stamps	BACnetARRAY[3] of	O[5]
Profile_Name	CharacterString	O

1 If one of these optional properties is present, then both of these properties shall be present.
2 If one of these optional properties is present, then both of these properties shall be present.
3 If one of these optional properties is present, then both of these properties shall be present.
4 This property is required if the object supports COV reporting.
5 These properties are required if the object supports intrinsic reporting.'

Multi-state Input Object Type
The Multi-state Output object type defines a standardized object whose properties represent the desired state of one or more physical outputs or processes within the BACnet Device in which the object resides. The actual functions associated with a specific state are a local matter and not specified by the protocol. For example, a particular state may represent the active/inactive condition of several physical outputs or perhaps the value of an analog output. The Present_Value property is

an unsigned integer number representing the state. The State_Text property associates a description with each state.
The Multi-state Output object type and its properties are summarized in Table 12-22 and described in detail in this subclause.

NOTE: Do not confuse the Present_Value state with the Event_State property, which reflects the offnormal state of the Multi- State Input.

Table 12-21. Properties of the Multi-state Input Object Type

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Present_Value	Unsigned	R[1]
Description	CharacterString	O
Device_Type	CharacterString	O
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	R
Reliability	BACnetReliability	O[2]
Out_Of_Service	BOOLEAN	R
Number_Of_States	Unsigned	R
State_Text	BACnetARRAY[N]of	O
Time_Delay	Unsigned	O[3]
Notification_Class	Unsigned	O[3]
Alarm_Values	List of Unsigned	O[3]
Fault_Values	List of Unsigned	O[3]
Event_E	BACnetEventTransitionBits	O[3]
Acked_Transitions	BACnetEventTransitionBits	O[3]
Notify_Type	BACnetNotifyType	O[3]
Event_Time_Stamps	BACnetARRAY[3] of	O[3]
Profile_Name	CharacterString	O

Multi-state Output Object Type

The Multi-state Output object type defines a standardized object whose properties represent the desired state of one or more physical outputs or processes within the BACnet Device in which the object resides. The actual functions associated with a specific state are a local matter and not specified by the protocol. For example, a particular state may represent the active/inactive condition of several physical outputs or perhaps the value of an analog output. The Present_Value property is an unsigned integer number representing the state. The State_Text property associates a description with each state.
The Multi-state Output object type and its properties are summarized in Table 12-22 and described in detail in this subclause.

Table 12-22. Properties of the Multi-state Output Object Type

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Present_Value	Unsigned	W
Description	CharacterString	O
Device_Type	CharacterString	O
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	R
Reliability	BACnetReliability	O
Out_Of_Service	BOOLEAN	R
Number_Of_States	Unsigned	R
State_Text	BACnetARRAY[N]of	O
Priority_Array	BACnetPriorityArray	R
Relinquish_Defaul	Unsigned	R
Time_Delay	Unsigned	O[1]
Notification_Class	Unsigned	O[1]
Feedback_Value	Unsigned	O[1]
Event_Enable	BACnetEventTransitionBits	O[1]
Acked_Transitions	BACnetEventTransitionBits	O[1]
Notify_Type	BACnetNotifyType	O[1]
Event_Time_Stam	BACnetARRAY[3] of	O[1]
Profile_Name	CharacterString	O

Multi-state Value Object Type

The Multi-state Value object type defines a standardized object whose properties represent the externally visible characteristics of a multi-state value. A "multi-state value" is a control system parameter residing in the memory of the BACnet Device. The actual functions associated with a specific state are a local matter and not specified by the protocol. For example, a particular state may represent the active/inactive condition of several physical inputs and outputs or perhaps the value of an analog input or output. The Present_Value property is an unsigned integer number representing the state. The State_Text property associates a description with each state.

The Multi-state Value object type and its properties are summarized in Table 12-23 and described in detail in this subclause.

NOTE: Do not confuse the Present_Value state with the Event_State property, which reflects the offnormal state of the Multi- state Value.

Table 12-23. Properties of the Multi-state Value Object Type

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Present_Value	Unsigned	R[1]
Description	CharacterString	O
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	R
Reliability	BACnetReliability	O[2]
Out_Of_Service	BOOLEAN	R
Number_Of_States	Unsigned	R
State_Text	BACnetARRAY[N] of	O
Priority_Array	BACnetPriorityArray	O[3]
Relinquish_Default	Unsigned	O[3]
Time_Delay	Unsigned	O[4]
Notification_Class	Unsigned	O[4]
Alarm_Values	List of Unsigned	O[4]
Fault_Values	List of Unsigned	O[4]
Event_Enable	BACnetEventTransition	O[4]
Acked_Transitions	BACnetEventTransition	O[4]
Notify_Type	BACnetNotifyType	O[4]
Event_Time_Stamps	BACnetARRAY[3] of	O[4]
Profile_Name	CharacterString	O

1 If Present_Value is commandable, then it is required to also be writable. This property is required to be writable when Out_Of_Service is TRUE.
2 This property shall be required if Fault_Values is present.
3 If Present_Value is commandable, then both of these properties shall be present.
4 These properties are required if the object supports intrinsic reporting.

Notification Class Object Type

The Notification Class object type defines a standardized object that represents and contains information required for the distribution of event notifications within BACnet systems. Notification Classes are useful for event-initiating objects that have identical needs in terms of how their notifications should be handled, what the destination(s) for their notifications should be, and how they should be acknowledged.

A notification class defines how event notifications shall be prioritized in their handling according to TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events; whether these categories of events require acknowledgment (nearly always by a human operator); and what destination devices or processes should receive notifications.

The purpose of prioritization is to provide a means to ensure that alarms or event notifications with critical time considerations are not unnecessarily delayed. The possible range of priorities is 0 - 255. A lower number indicates a higher priority. The priority and the Network Priority (Clause 6.2.2) are associated as defined in Table 13-5. Priorities may be assigned to TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events individually within a notification class.

The purpose of acknowledgment is to provide assurance that a notification has been acted upon by some other agent, rather than simply having been received correctly by another device. In most cases, acknowledgments come from human operators. TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events may, or may not, require individual acknowledgment within a notification class.

It is often necessary for event notifications to be sent to multiple destinations or to different destinations based on the time of day or day of week. Notification Classes may specify a list of destinations, each of which is qualified by time, day of week, and type of handling. A destination specifies a set of days of the week (Monday through Sunday) during which the destination is considered viable by the Notification Class object. In addition, each destination has a FromTime and ToTime, which specify a window, on those days of the week, during which the destination is viable. If an event that uses a Notification Class object occurs and the day is one of the days of the week that is valid for a given destination and the time is within the window specified in the destination, then the destination shall be sent a notification. Destinations may be further qualified, as applicable, by any combination of the three event transitions TO-OFFNORMAL, TO-FAULT, or TO-NORMAL.
The destination also defines the recipient device to receive the notification and a process within the device. Processes are identified by numeric handles that are only meaningful to the destination device. The administration of these handles is a local matter. The recipient device may be specified by either its unique Device Object_Identifier or its BACnetAddress. In the latter case, a specific node address, a multicast address, or a broadcast address may be used. The destination further specifies whether the notification shall be sent using a confirmed or unconfirmed event notification.
The Notification Class object and its properties are summarized in Table 12-24 and described in detail in this subclause.

Table 12-24. Properties of the Notification Class Object Type

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Description	CharacterString	O
Notification_Class	Unsigned	R
Priority	BACnetARRAY[3] of	R
Ack_Required	BACnetEventTransitionBi	R
Recipient_List	List of BACnetDestination	R
Profile_Name	CharacterString	O

Program Object Type
The Program object type defines a standardized object whose properties represent the externally visible characteristics of an application program. In this context, an application program is an abstract representation of a process within a BACnet Device, which is executing a particular body of instructions that act upon a particular collection of data structures. The logic that is embodied in these instructions and the form and content of these data structures are local matters.
The Program object provides a network-visible view of selected parameters of an application program in the form of properties of the Program object. Some of these properties are specified in the standard and exhibit a consistent behavior across different BACnet Devices. The operating state of the process that executes the application program may be viewed and controlled through these standardized properties, which are required for all Program objects. In addition to these standardized properties, a Program object may also provide vendor- specific properties. These vendor-specific properties may serve as inputs to the program, outputs from the program, or both. However, these vendor-specific properties may not be present at all. If any vendor-specific properties are present, the standard does not define what they are or how they work, as this is specific to the particular application program and vendor.
The Program object type and its standardized properties are summarized in Table 12-26 and described in detail in this subclause.

Table 12-26. Properties of the Program Object Type

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Program_State	BACnetProgramState	R
Program_Change	BACnetProgramRequest	W
Reason_For_Halt	BACnetProgramError	O[1]
Description_Of_Halt	CharacterString	O[1]
Program_Location	CharacterString	O
Description	CharacterString	O
Instance_Of	CharacterString	O
Status_Flags	BACnetStatusFlags	R
Reliability	BACnetReliability	O
Out_Of_Service	BOOLEAN	R
Profile_Name	CharacterString	O

1 If one of the optional properties Reason_For_Halt or Description_Of_Halt is present, then both of these properties shall be present.

Pulse Converter Object Type

The Pulse Converter object type defines a standardized object that represents a process whereby ongoing measurements made of some quantity, such as electric power or water or natural gas usage, and represented by pulses or counts, might be monitored over some time interval for applications such as peak load management, where it is necessary to make periodic measurements but where a precise accounting of every input pulse or count is not required.

The Pulse Converter object might represent a physical input. As an alternative, it might acquire the data from the Present_Value of an Accumulator object, representing an input in the same device as the Pulse Converter object. This linkage is illustrated by the dotted line in Figure 12-4. Every time the Present_Value property of the Accumulator object is incremented, the Count property of the Pulse Converter object is also incremented.

The Present_Value property of the Pulse Converter object can be adjusted at any time by writing to the Adjust_Value property, which causes the Count property to be adjusted, and the Present_Value recomputed from Count. In the illustration in Figure 12-4, the Count property of the Pulse Converter was adjusted down to 0 when the Total_Count of the Accumulator object had the value 0070. The object and its properties are summarized in Table 12-27 and described in detail in this subclause.

Table 12-27. Properties of the Pulse Converter Object

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Description	CharacterString	O
Present_Value	REAL	R[1]
Input_Reference	BACnetObjectPropertyRefere	O
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	R
Reliability	BACnetReliability	O
Out_Of_Service	BOOLEAN	R
Units	BACnetEngineeringUnits	R
Scale_Factor	REAL	R
Adjust_Value	REAL	W
Count	Unsigned	R
Update_Time	BACnetDateTime	R
Count_Change_Time	BACnetDateTime	R[2]
Count_Before_Chang	Unsigned	R[2]
COV_Increment	REAL	O[3]
COV_Period	Unsigned	O[3]
Notification_Class	Unsigned	O[4]
Time_Delay	Unsigned	O[4]
High_Limit	REAL	O[4]
Low_Limit	REAL	O[4]
Deadband	REAL	O[4]
Limit_Enable	BACnetLimitEnable	O[4]
Event_Enable	BACnetEventTransitionBits	O[4]
Acked_Transitions	BACnetEventTransitionBits	O[4]
Notify_Type	BACnetNotifyType	O[4]
Event_Time_Stamps	BACnetARRAY[3] of	O[4]
Profile_Name	CharacterString	O

1 This property is required to be writable when Out_Of_Service is TRUE.
2 These properties are required if Count_Before_Change is writable.
3 These properties are required if the object supports COV reporting.
4 These properties are required if the object supports intrinsic reporting.

Schedule Object Type
The Schedule object type defines a standardized object used to describe a periodic schedule that may recur during a range of dates, with optional exceptions at arbitrary times on arbitrary dates. The Schedule object also serves as a binding between these scheduled times and the writing of specified "values" to specific properties

of specific objects at those times. The Schedule object type and its properties are summarized in Table 12-28 and described in detail in this subclause.
Schedules are divided into days, of which there are two types: normal days within a week and exception days. Both types of days can specify scheduling events for either the full day or portions of a day, and a priority mechanism defines which scheduled event is in control at any giventime.
The current state of the Schedule object is represented by the value of its Present_Value property, which is normally calculated using the time/value pairs from the Weekly_Schedule and Exception_Schedule properties, with a default value for use when no schedules are in effect. Details of this calculation are provided in the description of the Present_Value property.
Versions of the Schedule object prior to Protocol_Revision 4 only support schedules that define an entire day, from midnight to midnight. For compatibility with these versions, this whole day behavior can be achieved by using a specific schedule format. Weekly_Schedule and Exception_Schedule values that begin at 00:00, and do not use any NULL values, will define schedules for the entire day. Property values in this format will produce the same results in all versions of the Schedule object.

Table 12-28. Properties of the Schedule Object Type

Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Present_Value	Any	R
Description	CharacterString	O
Effective_Period	BACnetDateRange	R
Weekly_Schedule	BACnetARRAY[7]of	O[1]
Exception_Schedule	BACnetARRAY[N]of	O[1]
Schedule_Default	Any	R
List_Of_Object_Proper	List of	R
Priority_For_Writing	Unsigned(1..16)	R
Status_Flags	BACnetStatusFlags	R
Reliability	BACnetReliability	R
Out_Of_Service	BOOLEAN	R
Profile_Name	CharacterString	O

1 At least one of these properties is required.

Trend Log Object Type
A Trend Log object monitors a property of a referenced object and, when predefined conditions are met, saves ("logs") the value of the property and a timestamp in an internal buffer for subsequent retrieval. The data may be logged periodically or upon a change of value. Errors that prevent the acquisition of the data, as well as changes in the status or operation of the logging process itself, are also recorded. Each timestamped buffer entry is called a trend log "record."
The referenced object may reside in the same device as the Trend Log object or in an external device. The referenced property's value may be recorded upon COV subscription or periodic poll. Where status flags are available (such as when the COVNotification or ReadPropertyMultiple services are used), they are also acquired and saved with the data.
Each Trend Log object maintains an internal, optionally fixed-size buffer. This buffer fills or grows as log records are added. If the buffer becomes full, the least recent record is overwritten when a new record is added, or collection may be set to stop. Trend Log records are transferred as BACnetLogRecords using the ReadRange service. The buffer may be cleared by writing a zero to the Record_Count property. Each record in the buffer has an implied SequenceNumber which is equal to the value the Total_Record_Count property has immediately after the record is added. If the Total_Record_Count is incremented past 2^{32}-1, then it shall reset to 1.
Several datatypes are defined for storage in the log records. The ability to store ANY datatypes is optional. Data stored in the log buffer may be optionally restricted in size to 32 bits, as in the case of bit strings, to facilitate implementation in devices with strict storage requirements.
Logging may be enabled and disabled through the Log_Enable property and at dates and times specified by the Start_Time and Stop_Time properties. Trend Log enabling and disabling is recorded in the log buffer.

Event reporting (notification) may be provided to facilitate automatic fetching of log records by processes on other devices such as fileservers. Support is provided for algorithmic reporting; optionally, intrinsic reporting may be provided.
In intrinsic reporting, when the number of records specified by the Notification_Threshold property have been collected since the previous notification (or startup), a new notification is sent to all subscribed devices. BUFFER_READY algorithmic reporting is described in Clause 13.3.7.
In response to a notification, subscribers may fetch all of the new records. If a subscriber needs to fetch all of the new records, it should use the 'By Sequence Number' form of the ReadRange service request.
A missed notification may be detected by a subscriber if the Current_Notify_Record it received in its previous notification is different than the Previous_Notify_Record parameter of the current notification. If the ReadRange-ACK response to the ReadRange request issued under these conditions has its 'first-item' flag set to TRUE, Trend Log records have probably been missed by this subscriber.
The acquisition of log records by remote devices has no effect upon the state of the Trend Log object itself. This allows completely independent, but properly sequential, access to its og records by all remote devices. Any remote device can independently update its records at anytime.

Table 12-29. Properties of the Trend Log Object Type

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Description	CharacterString	O
Log_Enable	BOOLEAN	W
Start_Time	BACnetDateTime	O[1,2]
Stop_Time	BACnetDateTime	O[1,2]
Log_DeviceObjectProp	BACnetDeviceObjectPropert	O[1]
Log_Interval	Unsigned	O[1,2]
COV_Resubscription_I	Unsigned	O
Client_COV_Incremen	BACnetClientCOV	O
Stop_When_Full	BOOLEAN	R
Buffer_Size	Unsigned32	R
Log_Buffer	List of BACnetLogRecord	R
Record_Count	Unsigned32	W
Total_Record_Count	Unsigned32	R
Notification_Threshold	Unsigned32	O[3]
Records_Since_Notific	Unsigned32	O[3]
Last_Notify_Record	Unsigned32	O[3]
Event_State	BACnetEventState	R
Notification_Class	Unsigned	O[3]
Event_Enable	BACnetEventTransitionBits	O[3]
Acked_Transitions	BACnetEventTransitionBits	O[3]
Notify_Type	BACnetNotifyType	O[3]
Event_Time_Stamps	BACnetARRAY[3] of	O[3]
Profile_Name	CharacterString	O

1 These properties are required to be present if the monitored property is a BACnet property.
2 If present, these properties are required to be writable.
3 These properties are required to be present if the object supports intrinsic reporting.

STANDARDISERAT UTFÖRANDE AV FLÖDESBILDER I SCADA

Allmänt

Standardiserat utförande skall ske enligt nedan.

BILDSTANDARD I SCADA

1. Bildritning

1.1 Symboler

1.1.2 Färger

Bilder ritas i följande färger:

Luftkanaler:		
Uteluft (tilluft ej beh.)	Blå	Färgkulör: 170
Tilluft	Röd	Färgkulör: 10
Frånluft	Gul	Färgkulör: 42
Avluft	Brun	Färgkulör: 36
Återluft	Orange	Färgkulör: 30
Rörledningar		
Fjärrvärme / primär värme	Lila	Färgkulör: 120
Huvudsekundär	Röd	Färgkulör: 10
Sekundär	Orange	Färgkulör: 30
Köldbärare	Blå	Färgkulör: 150
Köldmedium	Brun	Färgkulör: 34
Kylmedel	Grön	Färgkulör: 64
Kallvatten	Blå	Färgkulör: 160
Varmvatten	Grön	Färgkulör: 70
Övriga		Valfri enhetlig färg
Symboler och texter:		
Symboler och texter	Svart	Färgkulör 7

1:1:3 Linjer

Linjer ritas i följande tjocklekar:

Luftkanaler	Bredd: 10/10
Rörledningar	Bredd: 6/10
Rörledningar allmänt	Bredd: 4/10
Rörledningar ansl. sidoutr.	Bredd: 2/10

1:1:4 Texter

De dynamiska texterna (beteckningar m.m.) anpassas i storlek till symbolerna.

Komponenter ska vara försedda med system beteckning.

1:1:5 Disposition

Ett system / anläggning per bild

Det visade systemets informationstyngdpunkt ska finnas i mitten av bilden.
Bilden utformas så att systemets funktion framgår med största tydlighet.

Symboler som ritas utmed en linje ska vara väl centrerade utmed denna.

Texter och dynamik skall, då de placerats sida vid sida eller ovan / under, utgöra en rät linje i våg eller lod.

Då systemet inte får plats i sin helhet på en bild, indelas denna i två eller flera delar.

Delning utförs så att inga viktiga delar för systemet går förlorade.

Detta innebär att vissa delar av systemet måste förekomma på flera bilder.

LARM KLASSNING SAMT LARMUTRUSTNING

A-larm:	**C-larm:**	
Pumpgrop hög nivå	Givarfel	
Expansionskärlslarm	Övriga temperaturavvikelselarmer	
Pump förshunt driftstopp	Övriga driftslarm (verkningsgrad, servicelarmer m.m)	
Brandgasfläkt larm	Serviceomkopplare ej i auto	
Frysvaktslarm	Drifttidslarm	
Rökdetektorlarm	Handkörning lång tid av objekt	
Hisslarm nödsignal		
Instängningslarm	**Larmutskick till mobil via SCADA**	
Frysrum, fryskåpslarm	Arbetstid dagtid:	A-larm till respektive maskinist.
Kylrumslarm		Ej kvitterat inom en timme går a-larm till beredskapsmobil.
Kyl/frysapparatskåp A-summalarm		B- och C-larm rings ej ut.
Katastrofskydd	Övrig tid:	A-larm till beredskapsmobil.
Utlöst släckutrustning		B- och C-larm rings ej ut.
B-larm:		
Förshunt temperaturavvikelselarm		
Pump radkrets, vvc, ventilation, pumpgrop m.m driftstopp		
Fläktar driftstopp		
VVX summalarm		
Brand/brandgasspjäll fel läge		
Kyl/frysapparatskåp B-summalarm		
Tekniska fel (likriktare, fellarm inbrottslarm, hiss m.m)		
Övriga dvärgbrytare i apparatskåp		
Hög nivå fettavskiljare		

MÄTINSAMLING

ENERGIMÄTNING VVS

Följande system ska kunna mätas:
- Värme primär

Värde som ska kunna läsas i SCADA:
- Energi kWh
- Effekt kWh
- Effekt (momentant) kW
- Flöde m^3

Se även bilaga 1.1.

ENERGIMÄTNING EL

Följande system ska kunna mätas: se bilaga 1.1.
- Elförbrukning Apparatskåp
- Total fastighets el.

Värde som ska kunna läsas i SCADA:
- Energi kWh

Se även bilaga 1.1.

FLÖDESMÄTNING

Följande system ska kunna mätas:
- Kallvattenförbrukning,
- Varmvattenförbrukning ,

Värde som ska kunna läsas i SCADA:

- Flöde totalt m3
- Flöde momentant l/s

Se även bilaga 1.1.

RIKTLINJER FÖR IMPORT AV TIMVÄRDEN TILL MÄTINSAMLINGSPROGRAM MOMENTUM RC.	FTP / Mailimport : .txt .csv .xls .xlsx och liknande format. Annan importlösning Integrerad lösning via webbservice eller direkt integration mot SQL-databas. Filuppbyggnad: • Textfil med valfri kolumnavgränsare, (semikolon, komma, tab, fastbredd) • En rad per avläsning och endast en avläsning per rad. • Decimalavskiljare punkt (.) eller komma (,) Raduppbyggnad exempel: 1234E;2017-01-31 01:00;47.440 1234E;2017-01-31 02:00;47.600 Mätar-ID: Unikt värde för varje mätare (not 3) • Datum inklusive Klockslag: 2017-12-12 08:00 (not 4) visar energianvändningen mellan 07:00 och 08:00. Går att kompensera för system som använder sig av 08:00 = timme 8 (08:00-09:00) • Värde: Rekommenderar mätarställning, alternativt kan vi hantera förbrukningsvärden. **Not 1:** Mer data kan ingå i filen (exempelvis enhet) **Not 2:** Ordning på kolumner kan vara annan än den ovan **Not 3:** Unikt i Momentums Databas för varje mätare även vid exempelvis fjärrvärme måste Flöde och Värme ha separata ID. Dessa separeras fördelaktigt med suffix eller prefix till sitt ID alternativt kan enheten skickas med och slås samman med ett ID för anläggningen för att skapa ett unikt-ID. **Not 4:** Varje Fil måste innehålla kompletta dygn: Godkänt 1. Avläsning varje timme från 2017-01-31 01:00 till 2017-01-31 00:00 Godkänt 2. Avläsning varje timme från 2017-01-31 00:00 till 2017-01-31 23:00 (2) om 00:00 gäller avläsning för 00:00-01:00	