

Engineering Data Exchange Template for BACnet Systems



**“Description of the EDE Data Fields”
Version of Layout: 2.3**

Release Date: January 16, 2017

Purpose	4
Known limitations.....	4
General Format	5
File Naming conventions	5
Format Of The Sheet "EDE"	6
The Header	6
Project_Name.....	6
Version_Of_Reference_File	6
Time_Stamp_Of_Last_Change.....	6
Author_Of_Last_Change	6
Version_Of_Layout.....	6
Limited_Resources.....	6
The Data Point Columns.....	7
keyname	7
device-object-instance.....	7
object-name.....	7
object-type.....	7
object-instance	7
description	8
present-value-default.....	8
min-present-value.....	8
max-present-value.....	8
settable	8
supports COV	8
hi-limit	9
low-limit	9
state text reference.....	9
unit-code.....	9
vendor-specific-address	10
notification-class.....	10
Format of the Sheet "State-Texts"	11
Format of the Sheet "Unit-Texts"	11
Format of the Sheet "Object-Types"	11
Attention!.....	11

Document Revision History

Date	Author	Version	Revision information
14-06-2000	Frank Schubert MBS GmbH	1.0	Initial Release
21-06-2000	Frank Schubert MBS GmbH	1.1	
09-02-2004	N. Schmalstieg Siemens SBT	2.0	
19-12-2005	N. Schmalstieg Ing.-Büro Schmalstieg	2.1	Writable changed to Settable
09-01-2007	Ariën Peterse Regel Partners BV	2.2	Explained Settable
11-12-2015	Thomas Kurowski, Siemens Schweiz AG Frank Schubert, Beckhoff Automation	2.3	<p>Draft 1:</p> <ul style="list-style-type: none"> Added Notification-Class Added file naming conventions Harmonized Wording Updated example files Revised the "Units-Texts" sheet (incl. corrections for enumerations 189 and 190) and the corresponding description in the manual as per Addendum ar to 135-2012 Fixed the language for the definitions of min-present-value, max-present-value, hi-limit and low-limit while retaining the original meaning
03-05-2016	Thomas Kurowski, Siemens Schweiz AG		<p>Draft 2:</p> <p>Incorporated Feedback from BIG-EU WG-T and AMEV:</p> <ul style="list-style-type: none"> Added clarifying language on the #Limited_Resources field Harmonized wording (editorial changes)
01-10-2016	Thomas Kurowski, Siemens Schweiz AG		<p>Publication version:</p> <ul style="list-style-type: none"> translation of the manual to German minor editorial changes

Purpose

In a multi-vendor system engineering data need to be exchanged between the interacting parties. From the BACnet point of view it is not really required to have data point lists, because data of object properties from any device can be discovered online using the appropriate BACnet services. But depending on the BACnet functionality supported by server device or client device, it may be necessary to provide this information as an offline data-point list. This may also be required, if the client needs to be set-up before the server device is operational on the network.

The Engineering Data Exchange (EDE) template shall help to exchange engineering data, such as data point types, data point addresses and special data point presentation information in a standardized form.

When integrating multi-vendor systems using the BACnet protocol, the actual protocol is described in the BACnet Standard. The extent of implementation is described in the PICS (Protocol Implementation Conformance Statement) of a BACnet device.

BACnet objects contain various object properties that can help in setting-up the engineering data which may be required for the configuration of a Supervisory Station or an Automation Station. However, many of these object properties are not mandatory on either the client side or on the server side.

Also, many clients (Supervisory Stations or Automation Stations) require 'offline engineering' of their databases.

The Engineering Data Exchange template shall provide a common means to describe the objects of a data point server in the form of a Microsoft ® Excel spreadsheet.

Known limitations

Although the EDE file was never intended to provide a machine-to-machine data-exchange format it is commonly used to import or export information and exchange this information between controllers and workstations.

It is known that except from the object-name and object instance number only the usual properties of object types like analog, binary and multistate objects (input, output and value) can be exchanged using the EDE file. Detailed information for other applications like fire-alarm systems using life-safety objects or access control using the set of Access-Control objects, etc. is outside the scope of the current EDE-specification and will likely not be available using the EDE-file format described hereinafter.

General Format

The actual Excel workspace consists of four different sheets:

1. The sheet "EDE" contains project information and also the list of data points selected for data exchange.
2. The sheet "State-Texts" contains information about the state texts being used for binary objects and multi state objects. The sheet "EDE" refers or "points" to entries in the sheet "State Texts".
3. The sheet "Unit-Texts" contains a list of supported BACnetEngineeringUnits being used for analog objects. The sheet "EDE" refers or "points" to entries in the sheet "Unit-Texts". Proprietary units may be added if required, according to the rules defined in the BACnet Standard.
4. The sheet "Object-Types" contains a list of supported BACnet Object Types. The sheet "EDE" refers or "points" to entries in the sheet "Object-Types". Proprietary object-types may be added if required, according to the rules defined in the BACnet Standard.

Once the assignment between the keyname, object-name, object-type and object-instance has been communicated to the integration partner, this assignment shall not be changed between subsequent versions of the reference file. New objects should be added at the end of the sheet and should have instance numbers which have not been used for that device yet.

Comment lines may be included in the sheets. A comment line starts with the ANSI character '#'.

Additional optional columns may be added if necessary. The column headers have to be unique.

The XLS/XLSX spreadsheet file is mainly intended for human interaction. To allow for machine-to-machine data exchange in projects, the CSV (Comma Separated Value) file format is preferred.

The Excel version, storage format, country setting and separator are to be negotiated between all vendors for their common project.

File Naming conventions

The table-names in the XLS/XLSX-file shall be as follows:

EDE
State-Texts
Unit-Texts
Object-Types

The names of the CSV-files shall be assembled as follows:

Prefix:	Name agreed by the involved parties (e.g. the project name)	
Separator:	_ Underscore	
Suffix:	To uniquely identify the content, the suffixes shall be	
	EDE	-> identifies the main EDE table
	ObjTypes	-> identifies the object type table
	StateTexts	-> identifies the state texts used in Binary and Multistate objects
	Units	-> identifies the table of BACnet engineering units
Extension:	.csv	-> identifies the file type

Example Project-Name: "SunsetTower"

Excel-File:	"SunsetTower.xls" or "SunsetTower.xlsx"
EDE-File:	"SunsetTower_EDE.csv"
Table ObjectTypes:	"SunsetTower_ObjTypes.csv"
Table StateTexts:	"SunsetTower_StateTexts.csv"
Table Units:	"SunsetTower_Units.csv"

Issue: 16-01-2017

© Copyright 2017, BACnet Interest Group Europe

Format Of The Sheet “EDE”

The Header

Project Name

This field shall contain the project name and the location of the project.

Version Of Reference File

This field shall contain the version of the reference file. This actually means the version of the data point list.

Time Stamp Of Last Change

This is the date and time of the last change of this document.

Author Of Last Change

The full name of the author who did the last change to the document.

Version Of Layout

This is the version of the sheet layout that is used. The version of the layout should not change during a project. The description you are just reading is valid for layout version 2.3.

#Limited Resources

*This is an **optional comment field and as such not included** in the EDE template. If present, it should list limitations of the device such as:*

- *Maximum string length for keynames, descriptions and texts.*
- *Restrictions on address ranges (device object instances etc.)*
- *Special timing requirements*

The Data Point Columns

keyname

The keyname is the system wide unique name of the data object, as it will be displayed on the client's user interface.

In best case the keyname is identical to the object-name, but different names are allowed. Differing names may be necessary for example, if an operator wants to see a special structured keyname that is longer than the supported length of the object-name, or, in cases where there are many devices with identical object-names on the same network (e.g. AI1, BO1).

This field is mandatory.

device-object-instance

The device-object-instance identifies the device which contains the object described in this row. BACnet device object instances must be unique within all connected BACnet networks (the "BACnet internetwork"). If the project is a multi vendor system with different vendors, a range of device instances may be assigned to each of them by the responsible system integrator.

The device object instance is the lower 22 bits (bit 0 through 21) of the *Object_Identifier* property of the Device Object.

It is represented as a decimal value with a valid range from 0 – 4194302.

This field is mandatory.

object-name

This is the name of the object being described and identical to the *Object_Name property*. Object names shall be unique within a BACnet device. The object-name of the device-object itself shall be unique within the entire BACnet internetwork.

This field is mandatory.

object-type

This field contains a decimal value that represents the BACnetObjectType as used in the *Object_Type* property.

The sheet "Object-Types" contains the text descriptions for all supported object-types.

This information is part of the *Object_Identifier* property (most significant 10 bits i.e. bit 22 through 31).

This field is mandatory.

object-instance

This field contains the instance-number of the object as a decimal value. It covers the lower 22 bits (bit 0 through 21) of the *Object_Identifier* property. Within a BACnet-device different objects of the same object-type are distinguished by their instance numbers. The object instance therefore shall be unique for each object-type in a specific device.

The valid range is from 0 – 4194302.

This field is mandatory.

description

This information text provides more detailed description of the data point and its function.

It is often required for offline engineering.

The BACnet property *Description* is optional for a BACnet device, but should be identical to this entry, if present. Usually this text is taken from the VDI3814 point-list and is provided by the project design documents.

This field is optional.

present-value-default

The default value for the *Present_Value* property. If *Present_Value* is commandable, this field value shall be taken from the *Relinquish_Default* property.

This field is optional.

min-present-value

The minimum value that can be reliably obtained from or written to the *Present_Value* property.

If the *Min_Pres_Value* property exists in the referenced object, the value shall be taken from this property. If the *Min_Pres_Value* property does not exist in the referenced object, the value shall be specified by the allowed range of the application.

This field is optional.

max-present-value

The maximum value that can be reliably obtained from or written to the *Present_Value* property.

If the *Max_Pres_Value* property exists in the referenced object, the value shall be taken from this property. If the *Max_Pres_Value* property does not exist in the referenced object, the value shall be specified by the allowed range of the application.

This field is optional.

settable

This field indicates whether the writable *Present_Value* property is controlled by an automated process (device, Program) or can be set by a client.

The letter 'Y' means settable, the letter 'N' means not settable. Only properties which, by definition in the BACnet standard, are required to be writable or properties which can be prepared for write-access, can be settable. This is independent of whether write access is performed with or without priority management.

If the field is empty, the client will assume that the value may be read and set. If a write request is issued by a client, but not permitted by the server, the server will respond with the appropriate Error-PDU.

This field is optional.

supports COV

This field indicates whether the object supports COV or not. The letter 'Y' or an empty field indicates that the object supports COV. The letter "N" means COV is not supported.

This field is optional.

hi-limit

This is the upper alarm limit.

If the object supports intrinsic reporting and the *High_Limit* property is read-only, this field contains the value taken from the *High_Limit* property.

If the object supports intrinsic reporting and the *High_Limit* property is writable, this field shall be empty in order to indicate that the client may not rely on a fixed value as it may change during runtime.

If the object does not support intrinsic reporting, this field shall contain the value to be used for external monitoring for alarm conditions.

This field is optional.

low-limit

This is the lower alarm limit.

If the object supports intrinsic reporting and the *Low_Limit* property is read-only, this field contains the value taken from the *Low_Limit* property.

If the object supports intrinsic reporting and the *Low_Limit* property is writable, this field shall be empty in order to indicate that the client may not rely on a fixed value as it may change during runtime.

If the object does not support intrinsic reporting, this field shall contain the value to be used for external monitoring for alarm conditions.

This field is optional.

state text reference

This field is valid for binary and multi-state objects only.

The decimal values in this column are used as reference numbers referring to entries in the sheet "State-Text". Each entry within the "State-Text" sheet, identified by its reference number, offers an enumeration of state-texts which correspond directly to the possible values of the *Present_Value* property of the referencing object. Different objects listed in the "EDE" sheet may refer to the same entry in the "State-Text" sheet (use the same reference number) when their textual representations of *Present_Value* are identical.

This field is optional.

unit-code

This field is valid for analog and loop objects only.

The field contains a decimal value that represents the BACnetEngineeringUnits code taken from the corresponding *Units* property. In the "Unit-Text" sheet, the text descriptions for all supported unit codes can be found as human readable text.

This field is optional.

vendor-specific-address

This field may be used to identify addresses used in the server device (like terminal numbers). The address may provide an internal data-point-identification or reference.

This field is optional.

notification-class

This field contains the instance number of the notification_class object linked to the referenced object. If the object does not support intrinsic reporting, this field shall be empty.

This field is optional.

Format of the “State-Texts” sheet

This sheet consists of a number of rows, each identified by its reference number, listing an enumeration of state texts for multi-state objects (Text 1 ...Text n) or for binary objects (only Inactive-Text and Active-Text). Multi-State- and Binary-Texts shall not be mixed under one reference number i.e. in the same row. The row contains therefore either the list “Inactive-Text, Active-Text” or “Text1, Text2, Text3,...”.

The values of the reference numbers identifying the rows can be freely chosen by the creator of the document but shall be unique within this sheet.

For unused or not supported states the appropriate column is left empty. The “State-Text” sheet may contain more rows than actually referenced in a project to allow reuse of this sheet in other projects.

The “EDE” sheet refers to entries in the “State-Texts” sheet identified by their specific reference numbers. The “EDE” sheet and the “State-Texts” sheet must be maintained in a consistent manner.

Format of the “Unit-Texts” sheet

This sheet contains the unit texts for all BACnetEngineeringUnits codes and for all proprietary units being used in the project. The first column contains the unit code, the second column contains the associated text.

The “EDE” sheet refers to the unit code as indicated in the first column of the “Unit-Texts” sheet. Unit codes 0-255 and 47808-49999 match with the official ASHRAE unit-codes from the BACnet Standard. Unit codes in the ranges 256 - 47807 and 50000 - 65535 represent definitions of proprietary units. The “EDE” sheet and the “Unit-Texts” sheet must be maintained in a consistent manner (units referenced in the “EDE” sheet have to be present in “Unit-Texts” sheet).

Format of the “Object-Types” sheet

This sheet contains the object types for BACnet Standard objects and for all proprietary objects being used for interaction between different parties in this project. The first column contains the object-type code, the second column contains the associated text.

Object types supported but not listed in this sheet may be added on the vendor`s discretion.

The “EDE” sheet and the “Object-Types” sheet must be maintained in a consistent manner (for proprietary object-types).

Attention!

Numbers in the “Unit-Texts” and “Object-Types” sheets within the ASHRAE-reserved ranges shall not be changed. These codes are identical to the standard BACnet enumerations and may be used for automatic configuration purposes.