SLOVENSKI STANDARD
oSIST prEN 50090-6-3:2023

01-februar-2023

Stanovanjski in stavbni elektronski sistemi (HBES) - 6-3. del: Tretja zveza HBES Iot API

Home and Building Electronic Systems (HBES)- Part 6-3 -3rd Party HBES IoT API

Systèmes électroniques pour les foyers domestiques et les bâtiments (HBES) - Partie 6-3 : API tierce IdO HBES

Ta slovenski standard je istoveten z: **prEN 50090-6-3**

ICS:

35.240.67	Uporabniške rešitve IT v gradbeništvu	IT applications in building and construction industry
97.120	Avtomatske krmilne naprave za dom	Automatic controls for household use

oSIST prEN 50090-6-3:2023 **en**

EUROPEAN STANDARD

NORME EUROPÉENNE

EUROPÄISCHE NORM

DRAFT
prEN 50090-6-3

December 2022

ICS 35.240.67; 97.120 -

English Version

Home and Building Electronic Systems (HBES)- Part 6-3 -3rd Party HBES IoT API

Systèmes électroniques pour les foyers domestiques et les bâtiments (HBES) - Partie 6-3 : API tierce IdO HBES	To be completed

This draft European Standard is submitted to CENELEC members for enquiry.
Deadline for CENELEC: 2023-03-17.

It has been drawn up by CLC/TC 205.

If this draft becomes a European Standard, CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

This draft European Standard was established by CENELEC in three official versions (English, French, German).
A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Türkiye and the United Kingdom.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Warning : This document is not a European Standard. It is distributed for review and comments. It is subject to change without notice and shall not be referred to as a European Standard.

CENELEC
European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels

Project: 74475 Ref. No. prEN 50090-6-3 E

Contents

European foreword

This document (prEN 50090-6-3:2022) has been prepared by CLC/TC 205 “Home and Building Electronic Systems (HBES)”.

This document is currently submitted to the Enquiry.

The following dates are proposed:

• latest date by which the existence of this document has to be announced at national level	(doa)	dor + 6 months
• latest date by which this document has to be implemented at national level by publication of an identical national standard or by endorsement	(dop)	dor + 12 months
• latest date by which the national standards conflicting with this document have to be withdrawn	(dow)	dor + 36 months (to be confirmed or modified when voting)

1 Scope

This document defines a 3rd Party API for the Home and Building HBES Open Communication System.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

EN 50090-1:2011, *Home and Building Electronic Systems (HBES) - Part 1: Standardization structure*

EN 50090-3-3, *Home and Building Electronic Systems (HBES) - Part 3-3: Aspects of application - HBES Interworking model and common HBES data types*

EN 50090-6-2, *Home and Building Electronic Systems (HBES)- Part 6-2 IoT Semantic Ontology model description*

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in EN 50090-1:2011 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— IEC Electropedia: available at https://www.electropedia.org/

— ISO Online browsing platform: available at https://www.iso.org/obp

3.1.1
actuator
point performing an *actuation* (executed by a specific *procedure,* with an expected *result*) that changes an Installation state during Runtime

Note 1 to entry:

— The term Actuator can be mapped to sosa:Actuator in the SSN Ontology.

— The subject *actuation* can be mapped to sosa:Actuation in the SSN Ontology.

— The subject *procedure* can be mapped to sosa:Procedure in the SSN Ontology.

— The subject *result* can be mapped to sosa:Result in the SSN Ontology.

3.1.2
Application Function
function that uses a set of Functions to achieve the desired behaviour of a technical system, typically using a combination of devices exchanging information via their input and output Datapoints

Note 1 to entry: An Application Function may be split into several Functional Blocks with their input and output Datapoints that are logically connected to each other. The Functional Blocks may be located in one or more devices.

EXAMPLE Application Functions examples are “direct electrical heating”, “electrical heating with accumulators”, “warm water heating”, “fan coil air-conditioning” …

Note 2 to entry: The Application Function and Application are meant to be the same. Reason to introduce an alias term is to use a clear (understandable) reference from Application/ Application Function to the corresponding KIM class:ApplicationFunction or to the Function in the Management Client.

3.1.3
aspect

generally, a specific perspective on a system that contains things with different properties; a referencing mechanism to organize KIM elements in a specific perspective

EXAMPLE A Function Point is an ex officio Aspect with an important specific perspective. It is a referencing mechanism to organize together all to a Function Point interoperating Points (all GOs linked to a GA).

3.1.4
BIM

Building Information Model

digital process to describe and document a building in all its life cycle phases, from its planning, construction, operation up to its demolition

3.1.5
channel

collection of Datapoints of a device that are logically related to each other typically by association with a hardware feature or a specific function of that device

Note 1 to entry: These Datapoints may be derived from one or more defined Functional Blocks or may be an expansion above and beyond defined Functional Blocks or may be independent of a Functional Block if none is defined for the function associated with the Channel. The concept of a Channel is well-understood by the market participant, e.g. installers.

3.1.6
datapoint

represents a logical input entity of a device acting as recipient of Installation state data, whereas a logical output of a device acts as source of Installation state data

Note 1 to entry: In case of implementation as a Group Object, state data are communicated with the use of Function Points.

Note 2 to entry: The term Datapoint is the common term; to specifically denote a Datapoint available on an IoT 3rd Party API, the term IoT Datapoint is used.

3.1.7
device

physical element that is part of the network; it is a physical, concrete object that a customer can buy

3.1.8
endpoint

entry point to a service, a process, or a queue or topic destination in service-oriented architecture

3.1.9
Feature of Interest

abstraction of a real-world thing (phenomenon, equipment, person, event…) defined by its observable or actuatable properties

Note 1 to entry: In colloquial terms, a FOI is a property carrier.

Note 2 to entry: A Sensor operates on a FOI with observable properties, an Actuator with actuatable properties.

Note 3 to entry: A FOI is not a “classification/type” tag itself; the “classification/ type” is accomplished with the help of tags. Examples are defined in 4.5.1.4.

3.1.10
function
describes a part of the intended behaviour of a FB in a building context

3.1.11
Functional Block
consists of one or more Functions that belong together and that cannot be separated across two devices but big enough that a device with only one such Functional Block could be marketed

Note 1 to entry: A Functional Block has a well-defined black box behaviour.

3.1.12
Function Point
runtime system state information of a specific Application Function

Note 1 to entry: Shared by at least two Datapoints.

Note 2 to entry: Has a unique identifier that addresses a group of controlled objects. This identifier is called a Group Address.

EXAMPLE < Light Switch > in living room on/off, whereas the < … > is the Function Point name

3.1.13
Group Address
numerical identifier of a Function Point

3.1.14
Group Communication
communication model in which one sender communicates information to one and typically more receivers

Note 1 to entry: In IoT, this can be realized by simple UDP communication or by using a message broker system or other.

3.1.15
Group Object
foreseen for Group Communication using Group Address(es)

Note 1 to entry: May be accessed via point-to-point communication without an assigned Group Address. With assigned Group Address, it becomes a member of that Function Point represented by the Group Address.

3.1.16
HBES Information Model
ontology based model of HBES System relevant parts, including additional semantic (dictionary) information

Note 1 to entry: It is managed by the HBES Association, hence the abbreviation KIM.

3.1.17
Industry Foundation Classes
open standard to describe BIM data in a digital way

Note 1 to entry: IFC data and models are specified in ISO 16739-1.

3.1.18
installation
assembly of materials and components (devices) placed in position to provide a service

Note 1 to entry: An Installation is a deployed system (e.g. HVAC system or fire protection system) and consists of equipment and Functions that are used for a particular purpose.

Note 2 to entry: In relation to this term created data correlates to the installation model, described in 4.2.

[SOURCE: ISO 6707-1:2020, modified – added “(devices)” and Notes to entry.]

3.1.19
IoT Datapoint
represents an Endpoint at an IoT 3rd Party API that:

a) corresponds to one or more Function Points, such as a state data representation of a discrete state in a building context:

EXAMPLE 1 brightness → discrete state “brightness” is represented by the value 65 (percent)

b) is a fully qualified URL e.g. provided by an IoT 3rd Party Server

EXAMPLE 2 https://gateway.knx.local/knx/api/v1/datapoints/{Id}

3.1.20
IoT Function
represents a Function at an IoT 3rd Party API that:

— is as a collection of IoT Datapoints that fulfils a – by the user – intended behaviour

EXAMPLE “living room – rear light dimming”, “kitchen – floor heating”

Note 1 to entry: In a Mac, an IoT Function is instantiated data of a MaC Function in an Installation respectively MaC project. The MaC Function itself may base on an Application Function.

3.1.21
IoT 3rd Party API
set of requirements and regulations through which partial access to an Installation can be gained by offering a collection of Endpoints

3.1.22
IoT 3rd Party Client
device or service interacting with the Installation from outside using the IoT 3rd Party API

Note 1 to entry: The IoT 3rd Party Client connects to a single device that provides the IoT 3rd Party API and can use this single device to fully interact with the Installation, possibly depending on a specified authorization mechanism.

EXAMPLE 1 A mobile phone (from inside the network, or from an Internet connection) with typically short period connections.

EXAMPLE 2 A weather service permanently feeding in its weather information using the IoT 3rd Party API.

3.1.23
IoT 3rd Party Server
device that implements the IoT 3rd Party API

Note 1 to entry: This can be a dedicated device; this can be a function of a device that supports other HBES IoT and non HBES functionalities; it may be located within the local LAN of the IoT installation or outside.

3.1.24
MaC Catalog Entry
created management client data correlating to the product model, described in 4.2

3.1.25
MaC Function
Application Function created by the MaC and assigned to a building structure element, grouping several Group Addresses

3.1.26
MaC Project
project created by a MaC documenting the Configuration of an Installation

3.1.27
Management Client
means to configure and commission Devices as well as to plan, design and diagnose an entire Installation

Note 1 to entry: The MaC is used to configure and commission Devices, as well as to plan, design and diagnose an entire Installation. As a final step the MaC writes specific configuration data such as Device parameters to the Devices.

3.1.28
ontology
conceptual descriptions of things that have a real-world commonality sharing the knowledge of a domain, mainly expressed with OWL

Note 1 to entry: Ontologies are a structured way to describe the meaning of data in ontology classes and should not be mixed up with common data model structures.

3.1.29
Object Property
in OWL a built-in concept that **connects** pairs of individuals, an object property expression **represents** the (entire) relationship between the pairs of individuals

3.1.30
OWL
OWL 2 Web Ontology Language, informally OWL 2, specified by the World Wide Web Consortium (W3C), mainly serialized with XML syntax for RDF (RDF/XML)

Note 1 to entry: In this document the abbreviation OWL is always an explicit reference to OWL 2.

3.1.31
point
represents an interface to data in the system

Note 1 to entry: This document uses the term Point as an umbrella for data that can be accessed from outside of the Device, for instance to interact with other Points from other Devices. Consequently, term Point is a generic superset of the term Datapoint (which describes more precisely the technics how the “data” in the system are structured and/or coded).

3.1.32
Point API
simple RESTful (CoAP or HTTP) application programming interface designed for, but not limited to, constrained class 2 devices [RFC7228] supporting device individualization, device linking and accessing device runtime data (e.g. Functional Block or Channel Datapoints)

3.1.33
Quality Kind

represents a certain combination of observable or actuatable properties, available as predefined parts of the Semantic Dictionary or created individually during Configuration; the latter is the case when a Quality Kind with the intended combination of properties respectively tags is not (yet) part of the dictionary

Note 1 to entry: A QK is not a “classification/type” tag itself; the “classification/ type” is accomplished with the help of tags. Examples are defined in 4.5.1.4.

3.1.34
RDF

Resource Description Framework, as specified by the https://www.w3.org/RDF/

Note 1 to entry: RDF is a framework to represent information in the web by using triples. The information can be serialized and stored in many formats such as the TURTLE or JSON(-LD) format. The general RDF concept description can be found under https://www.w3.org/TR/rdf11-concepts/

3.1.35
runtime

process-to-process communication of data between devices, opposing to Configuration

Note 1 to entry: This concerns mainly the communication of Datapoint values (control and status information).

3.1.36
Semantic Export

project exported by the MaC reflecting an Installation in a linked data format

Note 1 to entry: The exported data are:

— structured according to the KIM, such as using Object Properties defined in KIM;

— annotated with additional semantic information from the Semantic Dictionary;

— referencing concepts of external Ontologies.

3.1.37
semantic dictionary

set of standardized terms allowing to annotate required parts of an Installation

Note 1 to entry: For details, see 4.2.8.

3.1.38
sensor

point performing an *observation* (executed by a specific *procedure,* triggered by a *stimulus*), responding a *result* as an Installation state during Runtime

Note 1 to entry:

— The term Sensor can be mapped to sosa:Sensor in the SSN Ontology.

— The subject observation can be mapped to sosa:Observation in the SSN Ontology.

— The subject stimulus can be mapped to ssn:Stimulus in the SSN Ontology.

— The subject *procedure* can be mapped to sosa:Procedure in the SSN Ontology.

— The subject *result* can be mapped to sosa:Result in the SSN Ontology.

3.1.39
tag
kind of annotation term used to extend available data with (in most cases) well known standardized information from a dictionary (in contrast to user defined, arbitrary term)

Note 1 to entry: A Tag is a concept-less term, without an integration in a broader concept such as the concept of a Datapoint (used in an Application Function), it has a limited semantic meaning.

EXAMPLE Term “flow” has a weak meaning on its own, but if you relate it in a FOI with the other term “water” this expresses at least that you observe/ actuate the water flow.

In this document a Tag is almost exclusively a term from the Semantic Dictionary.

3.1.40
thing description
semantic metadata model to describe (abstract or physical) things, as specified by the thing description https://www.w3.org/TR/wot-thing-description/ and thing Ontology https://www.w3.org/2019/wot/td

Note 1 to entry: TD relevant relations are described in the clause of Semantic Export.

3.2 Abbreviations

For the purposes of this document, the following abbreviations apply.

GA	Group Address
GO	Group Object
FB	Functional Block
FP	Function Point
HBES	Home and Building Electronic Systems
IFC	Industry Foundation Classes
IOO	Info On off
IO	Input Output
IoT	Internet of Things
KIM	KNX Information Model managed by KNX Association and described in EN 50090-6-2
KNXA	KNX Association
M	Mandatory
MaC	Management Client
OP	Object Property
O	Optional
TD	Thing Description

4 HBES IoT 3rd Party API

4.1 Introduction

4.1.1 Mapping of terms

The KNX Information Model expresses things that correspond to or reflect the wording and meaning used in the IoT world.

NOTE In the HBES community, the terminology has evolved and was developed over time having its origins more than 20 years ago. By making HBES IoT Installations accessible for customers via an open API, there is a need to intuitively adapt the language to make it understandable and usable by IoT developers. Unfortunately, it is not possible to preserve the historical HBES terminology and use it “as is“ when addressing external non-HBES developers.

The following Table 1 provides a mapping of terms used in the KNX Information Model, HBES IoT 3rd Party API and comparing it to the HBES-internal naming convention.

Table 1 — Mapping of terms

Customer *API Client Development*	**HBES IoT** *Server Development*	**KIM**	**HBES Classic** *Installer*
Function	IoT Function	*:ApplicationFunction*	Function
Datapoint0F	IoT Datapoint	*:FunctionPoint/:Datapoint*	Group Address/Group Object
building/floor/room/site	building/floor/room/site	*loc:Building/ loc:Floor/ loc:Room/ loc:Site*	Building/Floor/ Room/-
NA	NA	*:Channel*	Channel
Device	Device	*:Device*	Device

— The *Customer* terminology is relevant for an HBES IoT 3rd Party Client respectively for a developer building such a client based on the HBES IoT 3rd Party API. This developer is typically HBES agnostic.

— The HBES *IoT* and *KIM* terminology addresses the HBES manufacturer, building a HBES IoT 3rd Party Server, respectively is relevant for KNX Association itself specifying the Ontology.

— The HBES *Classic* terminology is well known for an installer of HBES S-Mode installations.

For a server development it is of essence to understand how to differentiate the terms from the table above from a technical perspective, for this see below.

Relations

HBES Classic makes use of the concept of a Group Address and Group Object: a **Group Address** links several **Group Objects** (a **Function** groups at least one **Group Address**, a **Device** has at least one **Group Object** assigned).

However, both a GA and GO are represented with a **Datapoint/ IoT Datapoint.**

Possibility to distinguish between a GA/ GO at the level of a **Datapoint**/ **IoT Datapoint,**

a) At level of the HBES IoT Server Development

Use of the corresponding KIM concepts for a GA or GO according the table above. Moreover, a Function Point also contains a property for the GA value.

b) **At the level of the API Client Development**

In the case where the server includes as part of the response the type information from the KIM concepts above and/or the optional property for the GA, with this it will be possible to distinguish a **Datapoint/IoT Datapoint** as being a GA or a GO in a KNX Classic Installation.

For details on how these relations are semantically expressed, refer to EN 50090-6-2.

4.1.2 General

The HBES IoT 3rd Party API shall support the following features:

- discovery of the HBES IoT 3rd Party Server
- reading and writing of End points
- setup notifications on End points
- access permission control of End points

Information about reliability and freshness of the Installation state data are not dealt with in this version of the HBES IoT 3rd Party API.

The HBES IoT 3rd Party API hosts in the HBES IoT 3rd Party Server the main server project and Installation concepts as depicted in Figure 1. The picture expresses the following aspects:

- with red dotted lines: the available relations in the API (which are fewer than the EN 50090 System or KIM describe, see explanation on D below)
- the item cardinality at the end of every relation (in UML notation)

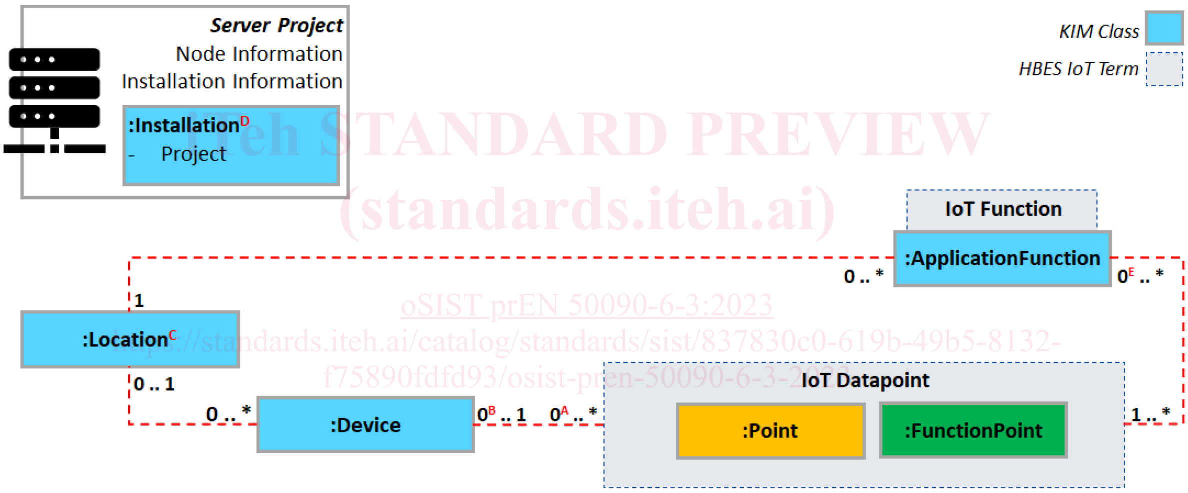

Figure 1 — Project elements in the HBES IoT 3rd Party Server

The letters in the above Figure denote the following:

A. when a Device does not support a Point (plain hardware like power supply, without any application) or when the IoT Datapoint represents a Function Point

B. when an IoT Datapoint represents a Function Point

C. includes floors/rooms/… and other

D. an Installation to location relation is not an explicit relationship in the API; it is expected that the resource path definitions of the category location (or similar) are always the known and fixed entry point for one (or more) Installations. In case of more installations, all content is merged, the API end points delivers a common perspective, see also 4.2.5.1.3.

E. when an IoT Datapoint represents a Point

Server Project - Node Information

Provides node meta information assigned to the server: geographically assigned location address, node (firmware) version and more

Server Project - Installation Information

Is the MaC project assigned to the server, in the form of the Semantic Export.

- Functions, Datapoints, Devices or Location elements such as a Building, Floor or Room.
- The version of the MaC Project, to track changes of the server project configuration. To identify a change the OP `:lastModified` of the `:Installation` can be used, see "Installation" in EN 50090-6-2).
- The HBES Information Model version (see the "Introduction" in EN 50090-6-2), as part of the Semantic Export.

NOTE Data and users are added to the actual server project in the HBES IoT 3rd Party Server in a plug-and-play way. Consequently, the server project is a superset/ extension of the original MaC project. Therefore, any additional data of the server project needs to be maintained by the HBES IoT 3rd Party Server and not by the MaC.

4.1.3 API - Content

The HBES IoT 3rd Party API consists of:

- Resource access methods (to set or retrieve Installation state data).
- End points hosting concepts comprising the Runtime communication or representing the Configuration of a (HBES Classic) Installation, additionally end points allowing to setup notifications on changes of Installation state data, provided to subscribers that are clients to the Installation.
- Methods to filter for specific resource items from a collection of resource items.
- Methods for authorization by an HBES IoT 3rd Party Client including a secure client/server communication.
- Access (permission) scopes, for security reasons the actual access to some end points (such as writing a Datapoint) is gated by the HBES IoT 3rd Party Server, this access will be granted as part of the authorization.

For all End points, their expected request/ response document formats, and their content is not described to the full extend in this document. Whether an element is mandatory or optional can be found in the electronic document (see 4.1.6). If an End point is mandatory or not, refer to 4.2.5.

NOTE Except by means of written text, Open API does not allow to define with a statement that an End point is optional.

4.1.4 API - Conformance

The HBES IoT 3rd Party API includes the above-described content and consists of the following:

- The API electronic document content is described with Open API (OAS3).
- The API concepts are described according JSON API, in case of deviations compared to the JSON API this is mentioned in this document. As a rule, an HBES IoT 3rd Party Server shall
  - ignore any valid JSON API content that is not supported or expected by the server;