

**BeStAddress User Guide**

**Part 1: INTRODUCTION & DATA**

Date: 20/11/2023

Version: 3.2

## Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Modified by | Modification |
| 1.0 | 06/05/19 | Gert De Jonge | First version |
| 1.7 | 29/11/2021 | Eddy Corthouts | Added section 6, Open data  Updated section 4.2, S350, “Address mutations file service”. Mutations are being reworked to achieve full harmonisation between the different regions.  A new mutations process is planned for release in Q2 2022 |
| 2.0 | 23/05/2022 | Luc Mertens | Adaptation to new services |
| 2.2 | 21/10/2022 | Eddy Corthouts | Added data element lengths to section 4.2 ”Entities”. Reviewed par. 2.1, “Data model” |
| 2.4. | 10/11/2022 | Eddy Corthouts | Split original user guide into 3 parts |
| 2.5 | 21/11/2022 | Luc Mertens | Adaptations after remarks partners |
| 2.6. | 14/12/2022 | Eddy Corthouts | Small changes to section 3 and spelling changes |
| 2.7. | 26/01/2023 | Eddy Corthouts | Section 3:  Removed “Figure 1, ‘BeSt Address conceptual model v23.3.1”.  Changed title of figure 2  Section 4.2.3, “postalInfo”: updated description |
| 2.8. | 02/05/2023 | Eddy Corthouts | Section 4.1.3: added explanation for status values |
| 2.9. | 02/05/2023 | Eddy Corthouts | Small corrections and enhancements; replaced figure 1, ‘Best Overview’ |
| 3.0 | 28/07/2023 | Evelyn Barreto | Update based on comments Mathieu Vanderstraeten (Brussels). |
| 3.1 | 24/08/2023 | Evelyn Barreto | Align the documentation with the changes that were made in the new XSD v24.3. |
| 3.2 | 20/11/2023 | Eddy Corthouts | Section 4.1.3, “Status”: added:   * status “rejected” for Flanders region. * In case the status is set to retired, no value is filled in in the validTo data element |

## Conventions

|  |  |
| --- | --- |
| Font | use |
| Italic | accentuation |

## Contact information

|  |  |
| --- | --- |
|  |  |
| Service Desk | [ServiceDesk.dto@bosa.fgov.be](mailto:ServiceDesk.dto@bosa.fgov.be)  +32 78 150312  +32 2 2129674 |

All BeSt services are subject to the latest BOSA FSB Terms & Conditions, such as described in [this document](file:///C:/Users/snichelson/Federale%20Horizontale%20Overheidsdiensten/OneDrive%20-%20Federale%20Horizontale%20Overheidsdiensten/BeST%20Business%20Analyse/WIP/fsb_governance_guidelines_fsb-user-guide-for-service-consumers%20NL%20v2.6.docx). The document describes the governance principles of the BOSA Service Bus as well.

Table of Contents

[Version History i](#_Toc151391341)

[Conventions i](#_Toc151391342)

[Contact information i](#_Toc151391343)

[Glossary 4](#_Toc151391344)

[1 Introduction 5](#_Toc151391345)

[1.1 Intended Audience 5](#_Toc151391346)

[1.2 Available documentation 5](#_Toc151391347)

[1.3 Purpose of this document 5](#_Toc151391348)

[2 BeSt application overview 6](#_Toc151391349)

[3 Data Model 9](#_Toc151391350)

[4 Entities and Data elements 12](#_Toc151391351)

[4.1 Support classes 12](#_Toc151391352)

[4.1.1 Enumerations 12](#_Toc151391353)

[4.1.2 Identifier 13](#_Toc151391354)

[4.1.3 Status 15](#_Toc151391355)

[4.1.4 GeographicalName 16](#_Toc151391356)

[4.1.5 GeographicalPosition 18](#_Toc151391357)

[4.2 Entities 20](#_Toc151391358)

[4.2.1 Municipality 20](#_Toc151391359)

[4.2.2 PartOfMunicpality 20](#_Toc151391360)

[4.2.3 PostalInfo 21](#_Toc151391361)

[4.2.4 StreetName 22](#_Toc151391362)

[4.2.5 Address 22](#_Toc151391363)

[4.2.6 AddressableObject 24](#_Toc151391364)

[4.2.7 History entities 24](#_Toc151391365)

[Document Information 25](#_Toc151391366)

[General 25](#_Toc151391367)

[Approbation 25](#_Toc151391368)

[Distribution 25](#_Toc151391369)

**List of Figures**

[Figure 1, ‘BeSt application Overview’ 6](#_Toc140242836)

[Figure 2, ‘BeSt Address conceptual data model - including history information’ 10](#_Toc140242837)

## Glossary

These are terms specific to this document, general terms known inside the BeSt environment are not added.

|  |  |
| --- | --- |
| Term | Description |
| Object | Object is a general term, it represents an independent element such as building, parcel, Address, Municipality, StreetName, PostalInfo… |
| Class | Template or blueprint that is used in modelling techniques to describes an object. |
| (BeSt) Identifier | Combination of the namespace, objectIdentifier and versionIdentifier which uniquely identify an object. |
| Entity | Representation of the BeSt object in the real world. The entity is identified by a complete BeStIdentifier. (So, every version of an object is an entity)  The entity corresponds with 1 record in our dataset. |
| Component | A sub part of an address or a StreetName. (Objects that are linked to another object)  Address has following sub parts: Municipality, StreetName, PostalInfo, PartOfMunicipality  A StreetName has 1 sub part: Municipality  Municipality, PostalInfo and PartOfMunicipality don’t have sub parts. |
| Linkable & Linked entity | Entities that have no components but are linked together because there is an address that defines this link. It concerns Municipality, PostalInfo and PartOfMunicipality.  Linked entities are always from another type considering 1 address can contain only 1 entity of each type. |
| History Chain | The history chain allows to retrieve the history of a particular Address (or address component), it consists of a chain of entities that make up the history of the address (or address component) |
| Parameter | (Input) item of the request interface. |
| Sub-parameter | Parameter that is part of a “combined” parameter. |
| Enumerated parameter | Parameter with a limited number of allowed values. This includes all Boolean parameters. |
| (output) Field | Output field in the reply interface of a service. |
| Prefix mun | For referrals to municipality in the name of a property, it is prefixed with mun |
| Prefix pom | For referrals to PartOfMunicipality in the name of a property, it is prefixed with pom |
| Prefix post | For referrals to PostalInfo in the name of a property, it is prefixed it with post |
| Prefix street | For referrals to StreetName in the name of a property, it is prefixed with street |
| Predecessor | The BeSt-identifier of the record that will be replaced. This will only be filled in on the ‘Add’ element |
| Successor | The BeSt-identifier of the record that is the replacement of the current record. This will only be filled in on the ‘Update’ element |
| MFT | Managed File Transfer |
| EventType | Type of change. This can be filled on or left blank. Each Region can have their own list of events. The Lists are added in the extensions. |
| SourceType | Indicates the original regional source of the address (Flanders, Brussels, Wallonia). |
| Correction | A cosmetic change or a spelling correction |

# Introduction

BeSt stands for “Belgian Streets”. The BeSt services provide address information on a federal level based on the three regional address master data sets from Brussels, Flanders and Wallonia.

BeSt was developed based on the agreement from 17 July 2019 between the Federal government, the Brussels region, the Flanders region and the Walloon region on how to reference and link address data.

In addition, the following organizations have participated in the development and implementation of BeSt:

* The National Geographic Institute (NGI)
* The General Administration of the Patrimony Documentation (AAPD) from the FPS Finance
* The National Registry (NR) from the FPS Internal Affairs
* Statistics Belgium from the FPS Economy
* The Crossroad Bank for Enterprises (CBE) from the FPS Economy
* The Directorate general Security and Prevention from the FPS Internal Affairs
* The FPS Governance and Support (BOSA)
* The Agency for Administrative Rationalization (DAV)
* The supplier of the universal postal services

## Intended Audience

This document is intended for any analyst or developer who wants to make use of the BoSa BeSt Address services.

## Available documentation

The next table provides an overview of the documentation available:

|  |  |
| --- | --- |
| **User guide** | **Purpose** |
| 1 BeSt\_Userguide\_INTRO\_and\_DATA  (This document) | Provides an overview of the BeSt application and describes the BeSt data, including the data model and the different data entities with their elements. |
| 2 BeSt\_Userguide\_WEB\_services | Describes the webservices that are available to the consumer to consult the BeSt address data |
| 3 BeSt\_Userguide\_MFT\_services | Describes the Managed File Services that are available to the consumer to obtain a full download file of BeSt addresses or to obtain daily mutations |

## Purpose of this document

This document is part of the service documentation and contains a general overview of the BeSt applications and describes the BeSt data, including the data model and the different data entities with their elements.

# BeSt application overview

Each region (Flemish, Brussels, Wallonia) has its’ own address master data set (“authentic source”) that contains the data to uniquely identify all address information in that region.

BOSA collects the data from the regional authentic sources to create a consolidated address database on federal level, the “BeSt address database”.



Figure 1, ‘BeSt application Overview’

**Obtaining the data from the regions**

The BeSt address database is initialised by loading it with a full download of the authentic sources of each region.

It is then kept up to date on a daily basis by applying mutations. The regions share this information in different ways:

* For Brussels and Wallonia, BoSa collects the daily mutations files directly from these regions
* Flanders offers a sync feed API that allows any user to follow the mutations on a permanent basis. For Flanders, BoSa consults the sync feeds, stores a local copy and uses that copy to transform the changes found in the feeds into the standard mutations file format.

This database includes historical data, starting from the start-up date of the mutations files.

**Making the data available to the consumer**

Bosa consolidates the address information from the 3 regions on a federal level and makes it available to the consumer in standard format.

The consumer can use different methods to access the information.

* MFT services
* Download weekly full download of address information

For the full download, Bosa collects the downloads from the 3 regions and puts them into 1 large full download (13 XML files), that is available to the consumer for download on a *weekly* basis.

* Download daily address Mutations

For the mutations, Bosa equally collects all the mutations from the 3 regions, consolidates them into 1 file and makes this file available for download on a *daily* basis.

* Download a weekly overview with structural anomalies.
* Webservices
* XML services

These are SOAP services retrieve their information *directly* from the regions by calling region APIs. These services are planned to be phased out once the new services are live.

* JSON services

These services retrieve their information from the BoSa BeSt address database and are REST services using JSON formatting.

**Keeping the data consistant**

BOSA verifies the consistency of the BeSt data on a regular basis: it compares the new full downloads with the content of the BOSA BeSt address database to reveal structural anomalies like missing foreign keys etc… These structural anomalies are then reported to the regions so they can correct them using the mutations files. Consumers can download these structural anomaly files to be informed.

In addition, any consumer can report an anomaly himself using the anomaly service. This service offers the consumer the possibility to create an anomaly report on a case-by-case basis, which is then emailed to the concerned region.

Note: as the BOSA address database is updated with mutations from the regions on a daily basis (every night), there may be a latency of 24 hours (in some cases 48hrs) in the data freshness.

# Data Model

This section describes the common model of address information (BeSt Model) that was agreed between the Federal government and the regions (Brussels, Flanders and Wallonia). It was established on the basis of the INSPIRE data model of the addresses. The BeSt Model is not compliant with the INSPIRE data model of the addresses, and therefore, with the EC Regulation 1089/2010.

Neverthless with BeSt model compliant data one can generate data that are compliant with both the INSPIRE data model of the addresses and the EC Regulation 1089/2010 (interoperability of spatial data sets and services).

As a matter of fact, BeSt can be considered as a “specialisation” of Inspire for the Belgian environment. Aspects where BeSt differs from Inspire include:

* Naming of certain data elements
* Use of separate classes for Municipality, partOfMunicipality, streetName, Address, postalInfo as opposed to one large class
* Omission of certain data elements that are qualified as “void” by Inspire

This approach was chosen for the sake of clarity for Belgian users and to be able to validate Belgian addresses based on the BeSt data model.

However, all data that are expected by Inspire – other than these qualified as “void” – are present in BeSt and can be converted to be delivered in the standard Inspire format.

The next diagram shows the conceptual BeSt data model (including history information). This model provides an overview of the different BeSt objects, their data elements and their relations.

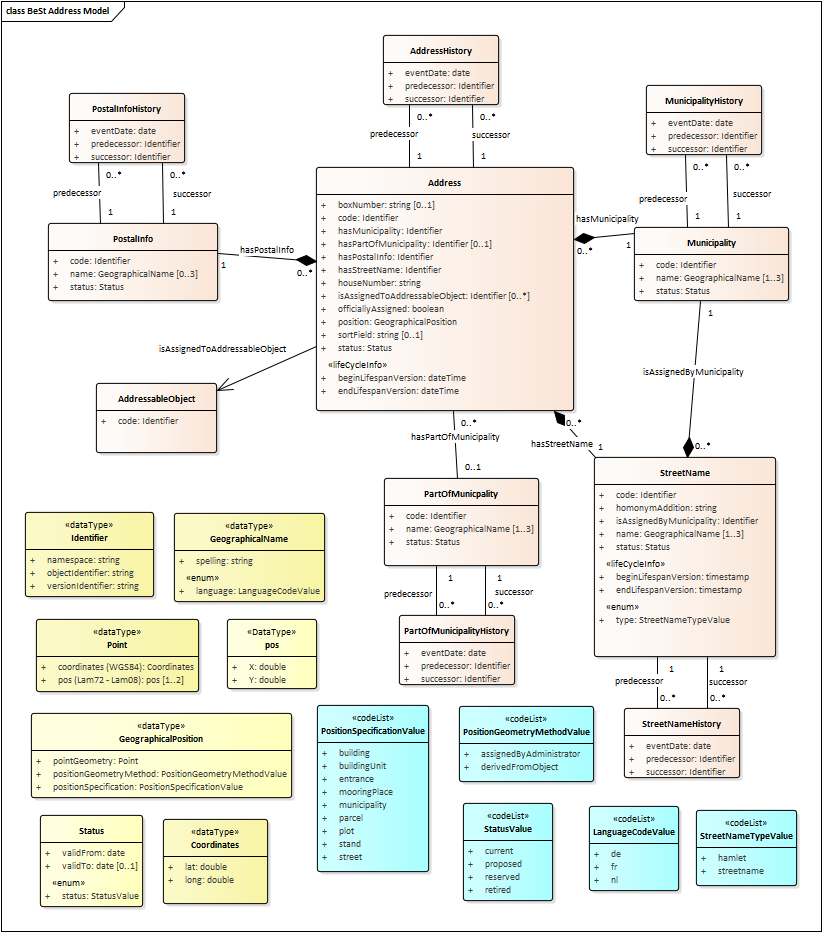


Figure 2, ‘BeSt Address conceptual data model - including history information’

The Address entity is the central entity in the model.

An address contains ‘components’ of other entities such as StreetName, a Municipality, possibly a PartOfMunicipality (only Wallonia) and a PostalInfo.

From the model, we can observe the following relations:

* Every address has a street
* Every address has a municipality
* Every address has a postal code
* Every street has a municipality

Some relations and attributes are “optional” (can have zero multiplicity). There is a logical explanation behind it:

* Wallonia works with PartOfMunicipality instead of PostalInfo
  + Every address in Wallonia has a PartOfMunicipality
  + The name of a PostalInfo is empty for Wallonia
* Other regios do not have a PartOfMunicipality
  + Their addresses have NO PartOfMunicipality
  + A postalInfo has always a name

Every entity has its’ own history table.

In the Inspire based model we can see that a logical relation in real life, the link between a postal code and a municipality, is not present. This means that we cannot derive a postal code from a municipality and vice versa unless an address exists that makes this link.

All the BOSA services make this link based on the Address Table.

In the next sections, the different objects of the model and their data elements are explained in detail.

# Entities and Data elements

This section described the BeSt entities and their data elements. It includes:

* The name of the subelement
* XSD definition used
* Format used in the BoSa database
* Cardinality of the subelement
* Description of the subelement

First, a number of support classes are introduced.

## Support classes

### Enumerations

Different enumerators are present:

* @ Status
  + - StatusValue
* @ GeographicalName
  + - LanguageCodeValue
* @ GeographicalPosition
  + - PositionSpecificationValue
    - PositionGeometryMethodValue
* @ StreetName
  + - StreetNameTypeValue

The exact list for each can be found where the containing element is explained.

BOSA considers the content of these attributes as a given value and will not store it as an enumeration value but as a string value. This allows us to capture values that might depart from the accepted list. The Bosa “compare” process will report such anomaly, but BOSA will not reject the record for this reason.

### Identifier

An Identifier is the key of a BeSt object and is composed of 3 subelements:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subelement** | **XSD data**  **type** | **Format BoSa DB** | **Cardinality** | **Description** |
| namespace | String | Varchar(55) | 1 | The namespace the regions uses to identify the region and the elment type |
| objectIdentifier | String | VARCHAR(20) | 1 | Actual object identifier |
| versionIdentifier | String | VARCHAR(30) | 1 | Version of the object. |

The “Identifier” uniquely identifies a StreetName, a Municipality, an Address,…

A versionIdentifier could be a datetimestamp (as is used by Wallonia and Flanders) or a number (as is used by Brussels).

Example of an address Identifier:

Namespace : [https ://data.vlaanderen.be/id/adres](https://data.vlaanderen.be/id/adres)

objectIdentifier : 12345

versionIdentifier : 18/04/2019 12:35

The persistence of a BeSt Identifier

The BeSt identifier is not persistent over time, it can change in certain situations. The regions are the owners of the address data and therefore decide the business logic concerning the persistence of an address Identifier. That business logic differs amongst the regions.

Note that when an Identifier changes, it can mean:

* the namespace changes
* the objectIdentifier changes
* the versionIdentifier changes

A namespace change will be rare. Namespaces are common for each type of object for each region. Any change requires careful planning.

VersionIdentifiers change frequently. Routinely, a small change[[1]](#footnote-2) to the data will result in a new versionIdentifier.

When a new version of an object is made, the old version gets a “validTo” date that is equal to the “validFrom” date of the new version.

ObjectIdentifier changes are reserved for “new” objects and big changes, for example when a street is renamed or a the house or box number of an address is changed.

Changes in adress component identifiers (address component identifiers are municipality, streetname and postalInfo) have also impact on objects that refer to them, especially with regards to versionIdentifiers.

In case the identifier of an address component changes, it probably results in a change of the address versionIdentifier considering the address contains a reference (foreign key) pointing to that component’s “old version” and now needs to reference the “new” version of that component.

Example:

a street gets a correction because of a spelling error in the name and therefore, the StreetName gets a new version. All addresses in that street should now point to the new StreetName. As a result, all these addresses would get a new version as well.

Unfortunately, this is not always the case. In Flanders, the version ID is not part of the unique key in their system. The BeSt identifier is considered as an attribute in their system. Changes to the versionIdentifier of a municipality do not lead to an update of a StreetName or an Address because their internal references (foreign keys) do not change. The BoSa application that translates the changes of Flanders[[2]](#footnote-3) into a mutations file also translates the keys of Flanders into BeSt identifiers. That process is aware of this problem and will also report an update for the streetNames and Addresses concerned without modifying their BeSt identifiers.

#### Relations

The model displays different relations. In version 23.3.1, these are not yet translated into attributes.

Version 24.2 did add these attributes[[3]](#footnote-4) explicitly to explain an interesting point: The relations in the model are ‘illustrative’.

What is meant here is that all references to other objects are made by mentioning the key (identifier).

In model design, it is common practice to translate a relation into an attribute of the object itself. (e.g. an address referring to a Municipality would be translated into an attribute ‘hasMunicipality’ in the object Municipality.)

In BeSt, the referring attribute is of the type “identifier’.

This is done to avoid that export files have to mention other types of objects if that is not foreseen in the definition.

Examples:

* A mutation file contains an updated address. All the components it refers to (Municipality, StreetName,…) are not changed and will be absent in the file.
* The full download is done by object (municipality, streetName, …). Each type can be found in a different file.

### Status

A status is more than an enumeration: it also contains the validity period.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subelement** | **XSD data**  **type** | **Format BoSa DB** | **Cardinality** | **Description** |
| status | StatusValue | Varchar(20) | 1 | The status is an enumerated attribute (Codelist StatusValue) with following values[[4]](#footnote-5).   * proposed * current * retired * reserved |
| validFrom | DateTime | TimeStamp | 1 | Is the startdate from which the entity is valid. |
| validTo | DateTime | TimeStamp | 0..1 | Indicates the end date of the validity of this entity.  Is empty when the entity is valid  A filled in validTo indicates that the entity is no longer active |

**Status values**

The status can have the following values:

* Reserved: the name of the street has been sent to the toponymy[[5]](#footnote-6) commission
* Proposed: the name has been accepted by the commission and is waiting for approval by the municipality council
* Rejected: the proposed name has been rejected by the municipality council
* Current: the municipality council has approved the name
* Retired: “retired” is only used in case an address or street is “physically” deleted, e.g. a street is converted to a parc. In case the status is set to retired, *no* value is filled in in the validTo data element.

*Status values by region*

Not every region uses every status value. The next table provides an overview of the use of status values by region:

|  |  |  |  |
| --- | --- | --- | --- |
| **Status value** | **BR** | **FL** | **WAL** |
| Reserved | - | - | - |
| Proposed | - | X | - |
| Rejected | - | X | - |
| Current | X | X | X |
| Retired | X | X | X |

*Status values by entity*

For the following entities, the status value is provided by the region:

* Address
* streetname
* partOfMunicipality (only used by Wallonia)

For Municipality and postalInfo, the status is not set by the regions as they are not the authentic source of the information.

For all municipalities, BOSA sets the status to current if that Municipality contains addresses with the status “current”

For partOfMunicipality and postalInfo, BOSA sets the status to “current” by default.

In the XSD, one will not find a “Status”, but an “addressStatus”, “municipalityStatus” etc….

This is done to be able to verify the status when the XML is created. But because the status for each object was not yet 100% certain at the outset of the project, it was decided to make independent status elements in the XSD. If one of the objects gets a different status, one has to change only that verification and not all the verifications. This has historically grown and is still present in the XSD.

**Validity period**

Directly related to the status is the validity period (validFrom – validTo). In case the status is missing, these subelements can be missing as well.

* validFrom
* validTo

When a new version of an object is made (same namespace and objectIdentifier), the validity of the old version ends, and the validity of the new version starts at the same moment.

The old version gets a “validTo” date that is equal to the “validFrom” date of the new version.

Although it is officially a date, in some region XML files the validity is shown as a timestamp. This has grown historically.

Note:

beginLifespanVersion and endLifespanVersion: these are technical timestamps that are only of interest to the owner of the data. They indicate when the record was created, respectively put out of service in the database of the owner.

### GeographicalName

Names like municipalities are multi-lingual in Belgium. A special class is used to express this multi-lingual situation.

The GeographicalName consist of the following subelements:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subelement** | **XSD data**  **type** | **Format BoSa DB** | **Cardinality** | **Description** |
| spelling | String | Varchar(100) | 1 | The real content of the name in the language specified |
| language | LanguageCodeValueType | Varchar(2) | 1 | Language in which the spelling is written. The language can have 3 values and is specified in the languageCodeValue.   * Nl * Fr * De   Every language is maximal once present for each object. |

In the database, the different language values are stored in 3 separate elements:

* nameNl Varchar(100)
* nameFr Varchar(100)
* nameDe Varchar(100)

### GeographicalPosition

This class is used to describe the position of an address.

The class uses other subclasses. The BoSa DB only stores the relevant attributes (which is why this document specifies a format for some attributes).

The GeographicalPosition consist of the following subelements:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subelement** | **XSD data**  **Type** | **Format BoSa DB** | **Cardinality** | **Description** |
| pointGeometry | Point |  | 1 | The location |
| positionGeometryMethod | PositionGeometryMethodValueType | VARCHAR(50) | 1 | enumerated value (codelist PositionGeometryMethodValue):   * assignedByAdministrator * derivedFromObject |
| positionSpecification | PositionSpecificationValueType | VARCHAR(50) | 1 | enumerated value (codelist PositionSpecificationValue):   * Building * buildingUnit * entrance * mooringPlace * municipality * parcel * plot * stand * street |

At this time, the point is noted in the lambert 72 system. Considering the future needs, the latest version can express more than 1 system (Requests have been made to express the location in the Lambert08 system).

The Point consist of the following subelements:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subelement** | **XSD data**  **Type** | **Format BoSa DB** | **Cardinality** | **Description** |
| pos | Pos  (DoubleList of double) | Element containing 2 decimals  Postgres definition: point GEOMETRY(POINT,31370). | 1..2 | This is the position expressed in lambert72 and the lambert08 notation  The lambert notation is a point with X and Y properties that are doubles (real). |
| coordinates | Coordinates | Element containing 2 decimals  Postgres definition: point GEOMETRY(POINT,31370). | 0..1 | Adding of the WGS84 standard (World Geodetic System 1984 using longitude and latitude) to the list of usable systems. |

In the XSD, the Pos and Coordinate are further specified. One of their attributes will indicate the system which is used to express the position. Currently, there are no fixed values decided for that attribute to indicate the used system. At this moment, this is not needed: they all use Lambert72.

Only the new webservices will return the position in all 3 formats and always in the same order: Lambert72, Lambert08, WGS84. Conversions are made using a converter software provided by the NGI (National Geographic Institute).

## Entities

This section describes the entities of the BeSt data model and their data elements.

For data elements that were discussed under support classes, only a reference is made to the support class.

### Municipality

A village or town (municipality) is the lowest level of government in Belgium. A municipality has the following attributes:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **XSD data**  **type** | **Format BoSa DB** | **Cardinality** | **Description** |
| code |  | See “Identifier” | 1 | Unique identifier; the objectIdentifier is currently the NIScode |
| name |  | See “GeographicalName” | 1..3 | Municipality name |
| status |  | See “  Status” | 1 | Municipality status |

### PartOfMunicpality

As is obvious from the name, this is a part of a Municipality. As mentioned before, only Wallonia uses the PartOfMunicipality.

The PartOfMunicipality has the following attributes:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subelement** | **XSD data**  **type** | **Format BoSa DB** | **Cardinality** | **Description** |
| code |  | See “Identifier” | 1 | Unique identifier; the objectIdentifier is a number (not the NIS code) |
| name |  | See “GeographicalName” | 1..3 | PartOfMunicipality name |
| status |  | See “  Status” | 1 | PartOfMunicipality status |

### PostalInfo

Set of information which, for a postal item, allows the unambiguous determination of an actual or potential delivery point. In most cases, the postal code coincides with the municipality, but this is not always the case.

Wallonia has no name for its postalInfo. Other regions will make a concatenation of all names of (part of) municipalities valid for the same postcode. For these regions, the name is present in at least 1 language, example: "Schaerbeek + Bruxelles (Pont Van Praet, Teichmann)".

The PostalInfo has the following attributes:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subelement** | **XSD data**  **type** | **Format BoSa DB** | **Cardinality** | **Description** |
| code |  | See “Identifier” | 1 | Unique identifier; the objectIdentifier is the postcode defined by Bpost. |
| name |  | See “GeographicalName” | 0..3 | Geographical zone that corresponds with the postal code (does not necessarily coincide with the municipality) |
| status |  | See “  Status” | 1 | PostalInfo status |

### StreetName

StreetName is usually unique within a municipality and is identified by a streetnamecode.

Occasionally, it can happen that one municipality contains two streets with the same name (e.g. after the merger of two municipalities). In Wallonia, such streets are distinguished through a different identifier, and they are linked to a different partOfMunicipality. In Flanders, such streets are distinguished through a different identifier and a homonym addition, for example: streetname Groenstraat (ANBO), Groenstraat (HOWI), where ANBO and HOWI are homonym additions.

The streetName has the following attributes:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subelement** | **XSD data**  **type** | **Format BoSa DB** | **Cardinality** | **Description** |
| code |  | See “Identifier” | 1 | Unique identifier |
| name |  | See “GeographicalName” | 1..3 | Street Name |
| status |  | See “  Status” | 1 | StreetName status |
| homonymAddition | CharStringType | VARCHAR (25) | 0..1 | Adding an homonym to the StreetName. |
| isAssignedByMunicipality | IdentifierType | IdentifierType | 1 | An Identifier that forms a link to the municipaIity where this street is located |
| type | StreetNameTypeValueType | VARCHAR (20) | 1 | Enumerated attribute. (codelist StreetNameTypeValue)   * hamlet[[6]](#footnote-7) * streetname |
| beginLifespanVersion | dateTime | Timestamp | 1 | Indicates when record was created in database of authentic source |
| endLifespanVersion | dateTime | Timestamp | 0..1 | Indicates when record was put of order in database of authentic source |

### Address

An address is a spatial object that in a human readable way identifies a fixed location of a property (building, parcel, …). For this purpose, an address has an identifier, e.g. an address number, which enables a user to distinguish it from the neighbor addresses, as well as a geographic position, which enables an application to locate the address spatially.

To identify the address unambiguously in a wider context, within the city, region and country, an address must be associated with a number of “address components” (municipality, streetName, partOfMunicipality) that define its location within a certain geographic area.

The address has the following attributes:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Attribute** | | **XSD data**  **type** | | **Format BoSa DB** | **Cardinality** | **Description** |
| code | |  | | See “Identifier” | 1 | Unique identifier |
| Status | |  | | See “  Status” | 1 | Address status |
| houseNumber | | CharStringType | | VARCHAR (15) | 1 | This is a string considering it might contain alphabetical values |
| boxNumber | | CharStringType | | VARCHAR (35) | 0..1 |  |
| sortField | | CharStringType | | VARCHAR (65) | 0..1 | Transformation of the house number and the box number (eg. By adding extra 0’s before) so this value can be sorted |
| officiallyAssigned | | boolean | | boolean | 1 | States if the address has been granted  officially |
| hasStreetName | | IdentifierType | | IdentifierType | 1 | BeStIdentifier that identifies the StreetName where this address is located. This attribute is always present |
| hasMunicipality | | IdentifierType | | IdentifierType | 1 | BeStIdentifier that identifies the Municipality where this address is located. This attribute is always present |
| hasPostalInfo | | IdentifierType | | IdentifierType | 1 | BeStIdentifier that identifies the postalInfo where this address is located. This attribute is always present |
| hasPartOfMunicipality | | IdentifierType | | IdentifierType | 0..1 | BeStIdentifier that identifies the partOfMunicipality where this address is located. This attribute is always present |
| isAssignedToAddressableObject | | (List of) IdentifierType with optional versionIdentifier | | (List of) IdentifierType with optional versionIdentifier | 0..\* | Reference to an addressable object to which the address is assigned. Currently only used in the mutations file. |
| Position |  | |  | | 1 | See “  GeographicalPosition”  This position is always present. At this moment, this is the lambert 72 position. The lambert notation is a point with X and Y properties that are doubles (real).  Bosa adds other standards to this position (using the NGI convertor) and offers also   * lambert08 notation * wgs84 notation (World Geodetic System 1984) using longitude and latitude to express the location. |
| beginLifespanVersion | dateTime | | Timestamp | | 1 | Indicates when record was created in database of authentic source |
| endLifespanVersion | dateTime | | Timestamp | | 0..1 | Indicates when record was put of order in database of authentic source |

### AddressableObject

Defined in the Inspire model, these are used to indicate whether an address is assigned to an addressable object (building, parcel…) or not. This information is very incomplete and can only be used for informational purposes.

Currently, only the mutations files contain references to addressable objects. (See also part 3 of the documentation)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **XSD data**  **type** | **Format BoSa DB** | **Cardinality** | **Description** |
| code |  |  | 1 | See “Identifier” |

### History entities

BOSA maintains a history entity for each of the BeSt entity. Each history table makes the link between 2 entities of the same kind that are historically linked.

Every history entity has the same structure:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subelement** | **XSD data**  **type** | **Format BoSa DB** | **Cardinality** | **Comment** |
| predecessor |  | See “Identifier” | 1 | Is the Identifier of the entity that is the predecessor |
| successor |  | See “Identifier” | 1 | Is the Identifier of the entity that is the successor |
| eventDate | dateTime | TimeStamp | 1 | date found in the first mutations file that announced this relation |

Since the principle is the same for all entities, this information is not repeated for every entity. (AddressHistory,PostalInfoHistory,MunicipalityHistory,StreetNameHistory,PartOfMunicipalityHistory)

One may ask why “predecessor” and “successor” are in a separate class and are not attributes of the main element classes. This choice was made because in the model v23.3, these attributes are not present. Hence, the decision was made to keep them separated.

# Document Information

## General

|  |  |
| --- | --- |
| Author(s) | Luc Mertens, Eddy Corthouts, Bart Hanssens |
| Document name: | BeStAddress Userguide INTRO and DATA |
|  |  |
| Number of pages: | 26 |
|  |  |
|  |  |
|  |  |
|  |  |

## Approbation

|  |  |  |
| --- | --- | --- |
| Nom | Fonction | Organisation |
| Johan Mertens | Service Manager | BOSA |
| François Soumillion | Integration Architect | BOSA |

1. Small changes are changes on the basis of which a human being can unambiguously conclude that the same thing is targeted (these can be spelling/typo/cosmetic corrections, but not material changes). [↑](#footnote-ref-2)
2. For Flanders, BOSA does not receive a mutations file but derives the mutations from the “sync feeds” that Flanders provides. [↑](#footnote-ref-3)
3. The attribute has the same name as the relation [↑](#footnote-ref-4)
4. For the moment, all object types uses the same status values. This might change in the future. [↑](#footnote-ref-5)
5. toponymy, taxonomic study of place-names, based on etymological, historical, and geographical information. [↑](#footnote-ref-6)
6. Hamlet: a small settlement, township, generally smaller than a village. [↑](#footnote-ref-7)