|  |  |
| --- | --- |
|  |  |

INTERFACE SPECIFICATION - ADDRESSMASTER

# Abstract

This document specifies the interfaces that are used between Ericsson Network Engineer (ENE) and Global Enterprise Service Bus (GESB). The purpose is to interface AddressMaster via GESB.

As both AddressMaster and Network Engineer can act in similar roles, the consequence is that the interface needs to be symmetric. Both system will be able to create, update, and delete address records. Reading/searching will be done from ENE to AM via GESB.

The technology used in the interface will be aligned with Telia architectural decisions. For simple reading, a synchronous technology is used, but for the other operations (create, update, delete) REST/json technology will be utilized.

Contents

[1 Introduction 4](#_Toc438018603)

[1.1 Purpose 4](#_Toc438018604)

[1.2 Scope 5](#_Toc438018605)

[1.3 Request and Responses 5](#_Toc438018606)

[1.4 REST Services 6](#_Toc438018607)

[2 Interface to ADDRESS MASTER 6](#_Toc438018608)

[2.1 DATA TYPES 6](#_Toc438018609)

[2.1.1 resHeaderType and publishSubscribeHeader 6](#_Toc438018610)

[2.1.2 eneAddressRecordType 7](#_Toc438018611)

[2.1.3 amAddressSourceIdType 9](#_Toc438018612)

[2.1.4 amFullAddressParamsType 10](#_Toc438018613)

[2.1.5 amFullAddressType 10](#_Toc438018614)

[2.1.6 amAliasAddressParamsType 13](#_Toc438018615)

[2.1.7 amAliasAddressType 13](#_Toc438018616)

[2.1.8 amBuildingParamsType 14](#_Toc438018617)

[2.1.9 amBuildingType 14](#_Toc438018618)

[2.1.10 amEntranceParamsType 14](#_Toc438018619)

[2.1.11 amEntranceType 15](#_Toc438018620)

[2.1.12 amPropertyType 15](#_Toc438018621)

[2.1.13 amPropertyAreaType 16](#_Toc438018622)

[2.1.14 amCoordinateType 17](#_Toc438018623)

[2.1.15 amWalkwayType 17](#_Toc438018624)

[2.1.16 amPolygonCoordinateParamsType 18](#_Toc438018625)

[2.1.17 amPolygonCoordinateType 18](#_Toc438018626)

[3 REST SERVICES GESB.ENE 18](#_Toc438018627)

[3.1 Create Services 19](#_Toc438018628)

[3.1.1 createEneAddress 19](#_Toc438018629)

[3.1.2 createEneWalkwayPath 19](#_Toc438018630)

[3.2 Update Services 20](#_Toc438018631)

[3.2.1 updateEneAddress 20](#_Toc438018632)

[3.2.2 updateEneWalkwayPath 20](#_Toc438018633)

[3.3 Delete Services 21](#_Toc438018634)

[3.3.1 deleteEneAddress 21](#_Toc438018635)

[3.4 Read/Search Services 21](#_Toc438018636)

[3.4.1 getAmFullAddress 21](#_Toc438018637)

[3.4.2 getAmBuilding 23](#_Toc438018638)

[3.4.3 getAmProperty 24](#_Toc438018639)

[3.4.4 getAmWalkway 25](#_Toc438018640)

[4 REST Services ENE.GESB 26](#_Toc438018641)

[4.1 Create Services 26](#_Toc438018642)

[4.1.1 createAmAddress 26](#_Toc438018643)

[4.1.1 createAmWalkwayPath 27](#_Toc438018644)

[4.2 Update Services 27](#_Toc438018645)

[4.2.1 updateAmAddress 27](#_Toc438018646)

[4.2.2 updateAmWalkwayPath 28](#_Toc438018647)

[4.3 Delete Services 28](#_Toc438018648)

[4.3.1 deleteAmAddress 28](#_Toc438018649)

[5 SOAP/XML Services ENE.ENE 29](#_Toc438018650)

[6 AddressMaster Response Code Handling 29](#_Toc438018651)

[7 GESB Target responseCode Handling 32](#_Toc438018652)

[8 REFERENCES 33](#_Toc438018653)

[9 APPENDIX A – AM-ENE Terminology Mapping 33](#_Toc438018654)

[10 APPENDIX B – Sample Messages 36](#_Toc438018655)

[10.1 Outgoing Messages – ENE as the client 36](#_Toc438018656)

[10.1.1 Create Example 36](#_Toc438018657)

[10.1.2 Search Example 37](#_Toc438018658)

[10.1.3 Update Example 38](#_Toc438018659)

[10.1.4 Delete Example 39](#_Toc438018660)

[10.2 Incoming Messages – ENE as the server 39](#_Toc438018661)

[10.3 Internal Messages – ENE as the client and server 39](#_Toc438018662)

[10.3.1 Search Example 40](#_Toc438018663)

[10.3.2 Create Example 40](#_Toc438018664)

[10.3.3 Update Example 40](#_Toc438018665)

[10.3.4 Delete Example 41](#_Toc438018666)

# Introduction

The document describes the functional specification for integrating the Telia‘s existing AddressMaster component with the target system Ericsson Integrated Inventory as part of the TAD-N replacement project. The services provided by GESB are Telia responsibility to implement. The services provided by ENE are implemented by Ericsson. Some service structures are proposed to be identical in the interface regardless of in which direction the service is used. This would include both the header and the parameter information of the json object.

For the purposes of internal communication to support edits within Network Engineer, the internal communication between the REST/json endpoints and the Network Engineer versioned database will be handled via SOAP/xml requests. These requests, though intended for internal use, may be directly used as well should latency appear to become an issue. At this time these are included for the purpose of documenting the potential use, but as of the authoring of this document are intended to be internal.

To manage information about the full Telia business system environment, Network Engineer needs a Telia internal identification number. This identifier will be decided by Telia, and it will be used in the interface.

N.B. To be able to read the tables correctly, it is recommended to change the viewing mode from “Print Layout” to “Draft” (selectable in   
View->Document Views). Print Layout corrupts some headings in conjunction with tables.

## Purpose

The purpose of this document is to outline the technical design of the interface between ENE and GESB in order to fulfill the functional needs provided by AddressMaster.

This document informs the target audience about the technology and the information pertaining to the communication protocol and error handling, which in turn provides a better understanding of the solution.

This document is intended for the following audiences:

* Telia and Skanova Solution Architects
* Telia and Skanova Implementation project team
* Ericsson Solution Architects
* Implementation owners for ESBs (MBA/GESB) and external systems

This document describes only logic related to interface communication itself. Address Management business logic is covered in a separated document: Ericsson Integrated Inventory Address Management HLD. See reference [2].

## Scope

The document describes the functional specification for integrating AddressMaster and Ericsson Integrated Inventory (EII, deployed over two applications, EGI and ENE) through an integration platform Global Enterprise Service Bus (GESB).

**EGI** is a comprehensive Service and Network Resource Management platform that stores and customizes all the physical and logical resources.

**ENE**, GIS-based network management system that helps you efficiently plan, design, maintain and document ongoing changes to your inside and outside plant facilities

**AdressMaster** is the master for address information in TS, with integrations to external databases. Other systems needing addresses can fetch them from the AddressMaster instead of maintaining the data within each system.

## Request and Responses

This document utilizes two datatypes that are specified in reference [3]. The datatypes are **reqHeaderType** and **resHeaderType**. These two header types are associated to request/response type of services.

The request header type is a common structure for representing and identifying messages exchanged by integrated systems.

The response header type is a common header used in every response. It contains unique ID for the message (“requestID”) and message outcome (OK, KO and in case error code and description). This header type uses a third datatype, namely **responseResultType** to return the individual response codes and potential additional information.

For AddressMaster integration, the response code and error messages are listed in chapter 3.2, even though the structure of the response codes can be found in reference [3].

## REST Services

Both ENE and GESB can play the role of both client and server due to that both parties can initiate operations that change the address record content.

REST endpoints will be used to facilitate the GESB sending information to ENE for the purpose of updating information. Address Master may receive updates from other third party applications and it is the responsibility of the GESB to inform ENE. The REST endpoint exposed by ENE will accept the updated address information and then internally update the ENE database.

The header type used for incoming update messages differs from the outgoing messag header. The datatype is called **publishSubscribeHeader**. The specification of the publishSubscribeHeader can be found in reference [3].

# Interface to ADDRESS MASTER

The source for the information is listed in ref 1. The Swedish translations of the attribute names can be found in a separate table at the end of the document. The translations are in the form of column names in a Swedish format, but without using the letters “å”, “ä”, and “ö”. The Example column in the tables may contain several examples.

The attribute names are written as they appear in the AddressMaster database schema, i.e. their column names. This mapping is for informational purposes only. The ENE attribute names are as defined in this interface document.

At any point in time in which the state of the address in the Network Engineer database is put into question (initial creation, update within ENE) a new attribute on the Address table will be toggled called AM\_VERIFIED. The purpose of this attribute is to ensure that any delays in data synchronization are available. Addresses that are not verified will not be sent to EGI.

## DATA TYPES

The data types defined by name in the following sections are presented in a tabular form for ease of reading. Please refer to Appendix B for sample message requests and responses using these datatypes.

### resHeaderType and publishSubscribeHeader

These two data types are utilized in the interface between ENE and AM. For the full specification of those data types, see reference [3].

### eneAddressRecordType

This data type mirrors the address record/address information that is stored in ENE. This record contains the most important and the most used attributes. Attributes outside this record are fetched from AddressMaster.

This data type is used both as an OUTPUT and INPUT data type in ENE. All values are optional and those filled in will be used by AddressMaster to filter the occurrences matching the criteria.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Description | Mandatory | Type | Example |
| pointId | Id for Address/Address detail | N | NUMBER | 100003711 |
| city | City/postal area | N | VARCHAR2(50 BYTE) | Stockholm |
| postalCode | Postal code/Zip code | N | VARCHAR2(5 BYTE) | 12581 |
| streetName | Street name | N | VARCHAR2(50 BYTE) | Mårbackavägen |
| streetCumber | Street number | N | VARCHAR2(5 BYTE) | 13 |
| entrance | Main entrance distinguished typically by a letter if several entrances | N | VARCHAR2(5 BYTE) | A |
| staircase | Staircase | N | VARCHAR2(10 BYTE) | ÖG |
| stairs | Floor number (BV being the lowest floor except for basement) | N | VARCHAR2(5 BYTE) | BV 4 |
| leftright | To the left/to the right (th and tv in Swedish) | N | VARCHAR2(2 BYTE) | TH TV |
| apartmentNumber | Apartment number | N | VARCHAR2(20 BYTE) | 1101 VILLA 125 X |
| long | X coordinate SWEREF99 | N | VARCHAR2(20 BYTE) | E18° 06 28.4400 E18°06.474 18.1079 |
| lat | Y coordinate SWEREF99 | N | VARCHAR2(20 BYTE) | N59°14 29.0399 N59°14.484 59.2414 |
| x\_coordinate | X coordinate RT90 | N | VARCHAR2(20 BYTE) | 6982797 |
| y\_coordinate | Y coordinate RT90 | N | VARCHAR2(20 BYTE) | 1544706 |
| z | Z coordinate indicating height in meters | N | VARCHAR2(20 BYTE) | 12 |
| aliasName | List of zero or many alias names | N | VARCHAR2(50 BYTE) | 100003711 |
| aliasTypeId | The identifying number for the type of alias. | N | NUMBER | 1 |
| amTypeIdDescription | The text value of the alias type id number. | N | VARCHAR2(100) | Populärnamn |
| propertyId | Property ID | N | VARCHAR2(9 BYTE) | 210067362 |
| municipality | Municipality is a Swedish administrative area. | N | VARCHAR2(16 BYTE) | SÖDERHAMN |
| district | Subarea of a municipality, e.g. a district/quarter or a farm. | N | VARCHAR2(40 BYTE) | ORSTA |
| block | A district can be divided into smaller blocks. This is an identifying number for a block. | N | VARCHAR2(4 BYTE) | 12 |
| blockUnit | A block can further be divided into smaller areas called units. This is an identifying number for a unit. | N | VARCHAR2(4 BYTE) | 16 |
| unitArea | A unit can further be divided into smaller areas called areas. This is an identifying number for an area. | N | NUMBER | 1 |
| propertyAreaX | List of **x** coordinates that together with similar y coordinates form pairs that form a polygon | N | VARCHAR2(20 BYTE) |  |
| propertyAreaY | List of **y** coordinates that together with similar x coordinates form pairs that form a polygon | N | VARCHAR2(20 BYTE) |  |
| entranceUuid | Official address ID connected to entrance | N | VARCHAR2(36 BYTE) | FA1FE7F4A79A140EE04073830F7348BB |
| buildingId | Building ID | N | NUMBER(8,0) | 125547140 |
| buildingType | Building Type code | N | VARCHAR2(3 BYTE) | 101 |
| buildingTypeName | The text value of the building type code | N | VARCHAR2(100 BYTE) | Kärnkraftverk |
| buildingStructureType | The type of the structure, indicating the kind of building category, e.g. SDU, MDU, SBU. | N | VARCHAR2(8 BYTE) | SBU |
| buildingUuid | Official address ID connected to building | N | VARCHAR2(36 BYTE) | 00EB46AA-BFF9-473D-AE46-7C857FFA83DB |
| walkwayPathText | Walkway path text | N | VARCHAR2(255 BYTE) | KB 43 |
| entranceCode | Entrance code for physical access | N | VARCHAR2(20 BYTE) | 1234 |
| addressType | The type or category of an address. One of LA/GA/CA/UA/WD. | N | VARCHAR2(2 BYTE) | WD |
| parentPointId | A local address and a coordinate address will point to a geographical address (by Point-ID) being the parent | Y | NUMBER | 100003711 |
| lmvFlag | This is a status flag indicating whether an address is an official government source, or whether lower level addresses are based on the official address. Official addresses are of type GA, and lower level addresses are of type LA. | Y | VARCHAR2(1 BYTE) | N  Y |

### amAddressSourceIdType

This is information about systems that are the source of attributes updates or any other operation where source system information is needed to be provided.

There is an originating source (ursprung\_kalla\_id) when the address is created and a source (kalla\_id) that shows witch source has done the latest update.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Description | Mandatory | Type | Example |
| amAddressSourceId | The identifying number of a system that is the latest source of the address information | N | NUMBER | An identification number the system which did the latest update. For ENE the number could be “137”. Decided by Telia. |
| amAddressNameId | The text value of the identifying number | N | VARCHAR2(50 BYTE) | SCB |
| amAddressNameIdDescription | Additional textual description supporting the amAddressNameId parameter | N | VARCHAR2(255 BYTE) | Statistiska Centralbyrån |
| amAddressOriginatingSourceName | The identifying number of a system that is the originating source of the address information | N | NUMBER | 137 |

### amFullAddressParamsType

This is a container of array of single entities. The single entities contain the full address information, which is defined as useful in ENE and EGI. This data type is used by AM/GESB to return data to ENE.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Description | Mandatory | Type | Example |
| <amFullAddressType> | Array entity of all alias address records, if any | N | array entities |  |

### amFullAddressType

This record contains all attributes from AddressMaster, of interest in ENE. This data type is used by AM/GESB to return data to ENE.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Description/Swedish name | Mandatory | Type | Example |
| pointId | Id for Address/Address detail | Y | NUMBER | 100003711 |
| city | City/postal area | N | VARCHAR2(50 BYTE) | Stockholm |
| postalCode | Postal code/Zip code | N | VARCHAR2(5 BYTE) | 12581 |
| streetName | Street name | N | VARCHAR2(50 BYTE) | Mårbackavägen |
| streetNumber | Street number | N | VARCHAR2(5 BYTE) | 13 |
| mainEntrance | Main entrance distinguished typically by a letter if several entrances | N | VARCHAR2(5 BYTE) | A |
| staircase | Staircase | N | VARCHAR2(10 BYTE) | ÖG |
| floor | Floor number (BV being the lowest floor except for basement) | N | VARCHAR2(5 BYTE) | BV or 4 |
| tltr | To the left/to the right (th and tv in Swedish) | N | VARCHAR2(2 BYTE) | TH, TV |
| apartmentNumber | Apartment number | N | VARCHAR2(20 BYTE) | 1101, VILLA, 125 X |
| typeId | Address level type code | N | NUMBER | 114 |
| typeIdName | The text value of the address level type code | N |  | Gatunamn |
| addressSourceId | Identification number for the source that did the latest address update | N | amAddressSourceIdType | See data type section |
| addressOriginalSourceId | Identification number for original address source | N | NUMBER | 13 |
| countryCode | Country code | N | VARCHAR2(3 BYTE) | SE |
| countryCodeName | Text value of the country code | N |  | SVERIGE |
| countyCode | County code | N | VARCHAR2(2 BYTE) | 14 |
| countyCodeName | Text value of the county code | N |  | HALLANDS LÄN |
| municipalityCode | Municipality code | N | VARCHAR2(2 BYTE) | 83 |
| municipalityCodeName | Text value of the municipality code |  |  | UPPLANDS VÄSBY |
| communityCode | Code for the community related to a religious administrative area | N | VARCHAR2(2 BYTE) | 19 |
| communityCodeName | Text value of the community code | N |  | KÅRSTA |
| amCoordinateParams | Identifier for a coordinate system (RT90, SWEREF99, WGS84) | N | amCoordinateType | R |
| aliasName | Identifier to a list of aliases | N | amAliasAddressParamsType | See data type section |
| property | Identifier to a specific property. | N | amPropertyType | See data type section |
| propertyAreaPolygon | List of coordinates outlines the borders of a property | N | amPolygonType | See data type section |
| propertyPolygonSourceId | Identification number for the source that did the latest polygon update | N | amAddressSourceIdType | See data type section |
| areaCode | The area code for the current address (address identified by the current pointId) | N | VARCHAR2(4 BYTE) | 08, 0961 |
| addressChangeFrom | Point\_ID to an address to be changed | N | NUMBER | 117362857 |
| addressChangeTo | Point\_ID to an address that will replace the changed address | N | NUMBER | 100214132 |
| changedAddressDate | Change date | N | DATE | 2006-03-07 |
| changedAddressSourceId | Identification number for the source that did the latest address update | N | amAddressSourceIdType | See data type section |
| entrances | All the entrances for all the buildings related to a specific address | N | amEntranceParamsType | See data type section |
| buildings | All the buildings related to a specific address | N | amBuildingParamsType | See data type section |
| siteObjectId | ID for object | N | VARCHAR2(8 BYTE) | 952933 |
| siteObjectTypeId | Site object type code | N | VARCHAR2(3 BYTE) | 116 |
| siteObjectTypeIdName | The textual value of object type code | N | VARCHAR2(50) | GALVANISK |
| siteSourceId | Identification number for the source of the site | N | amAddressSourceIdType | See data type section |
| walkwayPathText | Descriptive text to reach the address | N | VARCHAR2(255 BYTE) | KB 43 |
| walkwaySourceId | Identification number for the source of the walkway | N | amAddressSourceIdType | See data type section |
| walkwayObjectType | Walkway object type | N | VARCHAR2(20 BYTE) | ADRKOOR |
| entranceCode | Entrance code for physical access | N | VARCHAR2(20 BYTE) | 1234 |
| walkwayRegistrationDate | Date when walkway was updated | N | DATE | 2009-11-27 |

### amAliasAddressParamsType

This is a container of array of single alias entities. This data type is used by AM/GESB to return data to ENE.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Description | Mandatory | Type | Example |
| <amAliasAddressType> | Array entity of all alias address records, if any | N | array entities |  |

### amAliasAddressType

A single alias entity. A name can be an alias of different kind. Street names, places, apartment numbers; all these can be aliases, or have other names as related aliases. Alias names can be several.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Description | Mandatory | Type | Example |
| amAliasName | The alias address in text format | N | VARCHAR2(50 BYTE) | EKOLLONGR |
| amTypeId | The type of alias address | N | NUMBER | 1 |
| amAliasTypeIdName | Text value of the alias type | N | VARCHAR2(50 BYTE) | Stavningssynonym |
| amAliasAddressSourceId | An information record containing source information of aliases | N | amAddressSourceIdType | See data type section |

### amBuildingParamsType

This is a container of array of single alias entities. This data type is used by AM/GESB to return data to ENE.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Description | Mandatory | Type | Example |
| <amBuildingType> | Array entity of all alias address records, if any | N | array entities |  |

### amBuildingType

Information related to a building. This data type is used by AM/GESB to return data to ENE.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Description | Mandatory | Type | Example |
| buildingId | Identity of the building | N | NUMBER(8,0) | 10000305 |
| buildingType | Building type code | N | VARCHAR2(3 BYTE) | 101 |
| buildingTypeName | The text value of the building type code | N | VARCHAR2(100 BYTE) | Kärnkraftverk |
| habitationType | Habitation type | N | VARCHAR2(1 BYTE) | 1 |
| buildingSourceId | Identification number for the source of the building | N | amAddressSourceIdType | See data type section |
| buildingUuid | Official Address Id connected to Building | N | VARCHAR2(36 BYTE) | 00EB46AA-BFF9-473D-AE46-7C857FFA83DB |
| buildingStatus | Building status code | N | VARCHAR2(2 BYTE) | 1 |
| buildingStatusDescription | Text value of the building status code | N | VARCHAR2(200 BYTE) | Gällande under utredning |
| buildingCategory | The type of the structure indicating the building category as defined in Nätfråga. | N | VARCHAR2(3 BYTE) | MBU |

### amEntranceParamsType

This is a container of array of single alias entities. This data type is used by AM/GESB to return data to ENE.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Description | Mandatory | Type | Example |
| <amEntranceType> | Array entity of all alias address records, if any | N | array entities |  |

### amEntranceType

Information related to an entrance of a building. This data type is used by AM/GESB to return data to ENE.

For an entrance, return also the related building information.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Description | Mandatory | Type | Example |
| entranceId | The official identity for an entrance | N | VARCHAR2(32 BYTE) | FA1FE7F4A79A140EE04073830F7348BB |
| entranceType | Entrance type | N | VARCHAR2(1 BYTE) | 2 |
| entranceTypeDescription | Text value of the entrance type code | N | VARCHAR2(300 BYTE) | Bostadsadress med registrerad lägenhet |
| relatedBuilding | Building information to which the entrance belongs | N | amBuildingType | See data type section |
| entranceSourceId | Identification number for the source of the entrance | N | amAddressSourceIdType | See data type section |
| entranceStatusCode | Entrance status code | N | VARCHAR2(2 BYTE) | 1 |
| entranceStatusDescription | Text value of the entrance status code | N | VARCHAR2(200 BYTE) | Planerad |
| elevatorStatusCode | Elevator status code | N | VARCHAR2(50 BYTE) | 9 |
| elevatorStatusDescription | Text value of the elevator status code | N | NVARCHAR2(50 CHAR) | Ja |
| entranceUuid | Official address ID connected to entrance | N | VARCHAR2(36 BYTE) | 4952F12F-CADD-4155-9226-B047AF9C1C43 |

### amPropertyType

Information related to a property. This data type is used by AM/GESB to return data to ENE.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Description | Mandatory | Type | Example |
| propertyId | Property ID | N | VARCHAR2(9 BYTE) | 210067362 |
| propertyMunicipality | Municipality (part of property name) | N | VARCHAR2(16 BYTE) | SÖDERHAMN |
| propertyQuarter | Quarter/District (part of property name) | N | VARCHAR2(40 BYTE) | ORSTA |
| propertyBlock | Main Number (part of Property name) | N | VARCHAR2(4 BYTE) | 12 |
| colonChr | Colon character (part of property name) | N | VARCHAR2(1 BYTE) | : |
| propertyUnit | Sub number (part of property name | N | VARCHAR2(4 BYTE) | 16 |
| areaNumber | Area number (part of property name | N | NUMBER | 1 |
| propertyX | X coordinate, center point for property in SWEREF 99 format- Conversion done by AM | N | VARCHAR2(20 BYTE) | 6796711 |
| propertyY | Y coordinate, center point for property in SWEREF99 format. Conversion done by AM | N | VARCHAR2(20 BYTE) | 1561999 |
| propertySourceId | Identification number for the source that did the latest property update | N | amAddressSourceIdType | See data type section |

### amPropertyAreaType

Information related to a property area, i.e. its polygon properties. This data type is used by AM/GESB to return data to ENE.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Description | Mandatory | Type | Example |
| propertyAreaPolygon | List of coordinates outlines the borders of a property | N | amPolygonCoordinateParamsType | See data type section |
| propertyPolygonSourceId | Identification number for the source that did the latest polygon update | N | amAddressSourceIdType | See data type section |

### amCoordinateType

Information related to specific coordinates. This data type is used by AM/GESB to return data to ENE.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Description | Mandatory | Type | Example |
| coordinateSystemIdName | Text value of the coordinate system | N | VARCHAR2(255 BYTE) | RT 90 2,5 gon V |
| coordinateTypeId | Coordinate type identity (R,E,B,I,T,U,Z,M,A,V,1,2,3,4,5) | N | VARCHAR2(20 BYTE) | V |
| coordinateTypeIdName | Text value of the coordinate type identity | N | VARCHAR2(255 BYTE) | Verifierad via ENE |
| long | X coordinate in SWEREF99 | N | VARCHAR2(20 BYTE) | E18° 06 28.4400 E18°06.474 18.1079 |
| lat | Y coordinate in SWEREF99 | N | VARCHAR2(20 BYTE) | N59°14 29.0399 N59°14.484 59.2414 |
| z | Z coordinate. Height in meters. | N | VARCHAR2(4 BYTE) | 4 |
| x\_coord | X coordinate in RT90. AM supplies the RT90 coordinates. | N | VARCHAR2(20 BYTE) | 6982797 |
| y\_coord | X coordinate in RT90. AM supplies the RT90 coordinates. | N | VARCHAR2(20 BYTE) | 1544706 |
| coordinateSourceId | Identification number for the source that did the latest coordinate update | N | amAddressSourceIdType | See data type section |

### amWalkwayType

Information related to a walkway path. This data type is used by AM/GESB to return data to ENE.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Description | Mandatory | Type | Example |
| walkwayPathText | Descriptive text to reach the address | N | VARCHAR2(255 BYTE) | KB 43 |
| walkwaySourceId | Identification number for the source of the walkway | N | amAddressSourceIdType | See data type section |
| walkwayObjectType | Walkway object type | N | VARCHAR2(20 BYTE) | ADRKOOR |
| entranceCode | Entrance code for physical access | N | VARCHAR2(20 BYTE) | 1234 |
| walkwayRegistrationDate | Date when walkway was updated | N | DATE | 2009-11-27 |

### amPolygonCoordinateParamsType

This data type is of dynamic size and contains an array of amPolygonCoordinateTypes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Description | Mandatory | Type | Example |
| <amPolygonCoordinateType> | Dynamic array entity of X/Y coordinate pairs | N | array entities |  |

### amPolygonCoordinateType

This data type is an entity containing coordinate information. The coordinate information is used to create full polygon information.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Description | Mandatory | Type | Example |
| polygonX | SWEREF99 format | Y | VARCHAR2(20 BYTE) | E18° 06 28.4400 E18°06.474 18.1079 |
| polygonY | SWEREF99 format | Y | VARCHAR2(20 BYTE) | N59°14 29.0399 N59°14.484 59.2414 |

# REST SERVICES GESB.ENE

The services in this section are those that are provided from GESB, i.e. GESB is the server (or provider) and ENE is the client (or consumer).

## Create Services

### createEneAddress

Create a new Local Address or on Apartment Number level (could also be a SDU or Business CAP) for a Customer Site or Coordinate Address for a Network object if it is missing in AM.

When this call is being made as a result of ENE, it is the responsibility of the GESB to make an additional call back into ENE using the appropriate REST service in order to provide ENE with the newly assigned punkt\_id. At this time, ENE will mark the address as verified.

***Message***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Mandatory | Type | Description | Example |
| pubSubHeader | yes | publishSubscribeHeader | Message header | See data type section |
| addressRequestParams | yes | eneAddressRecordType | Based on what type of address is created, different subsets of attributes in this record are filled in. |  |

### createEneWalkwayPath

When a walkway path is needed in ENE-EGI it shall be able to create in AM to get PointID and to connect to Address Point ID or other Network Object without PointID. AM shall store info about the type of object it is (ex FP, SP).

***Message***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Mandatory | Type | Description | Example |
| pubSubHeader | yes | publishSubscribeHeader | Message header | See data type section |
| pointId | no | NUMBER | An existing Point-Id, or empty value if a new Point-Id is to be created. | 100003711 |
| walkwayParams | yes | amWalkwayType | The walkwayObjectType attribute needs to be aligned with AM. The typical usage is that an ENE user creates walkway paths. | See data type section |

## Update Services

### updateEneAddress

Update of a Local Address on Floor or Apartment Number level (could also be a SDU or Business CAP) for a Customer Site or Coordinate Address for a Network object that ENE-EGI is using.

***Message 1***

The update request is performed on a Point-Id, and the attributes to be updated, potentially a subset, are in the addressRequestParams parameter.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Mandatory | Type | Description | Example |
| pubSubHeader | yes | publishSubscribeHeader | Message header | See data type section |
| pointId | yes | NUMBER | Point-Id of the requested address | 100003711 |
| addressRequestParams | yes | eneAddressRecordType | Attributes and attribute information that are sent to AM for an update. |  |

### updateEneWalkwayPath

Update of information related to a walkway path. This update is done in AM and the updated information is published to ENE.

***Message***

The update request is performed on a Point-Id, and the attributes to be updated, potentially a subset, are in the addressRequestParams parameter.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Mandatory | Type | Description | Example |
| pubSubHeader | yes | publishSubscribeHeader | Message header | See data type section |
| pointId | yes | NUMBER | Point-Id of the walkway path to be updated | 100003711 |
| walkwayParams | yes | amWalkwayType | This structure contains the new parameters to be updated. | See data type section |

## Delete Services

### deleteEneAddress

Only delete the connection between PointID and ENE-EGI object of the subscription of the PointID. The message will not trigger an actual delete in AM.

***Message***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Mandatory | Type | Description | Example |
| pubSubHeader | yes | publishSubscribeHeader | Message header | See data type section |
| pointId | yes | NUMBER | Point-Id of the address to be deleted | 100003711 |

## Read/Search Services

### getAmFullAddress

This service makes it possible to search for existing information in AM based on different input data. As ENE only stores a subset of valid address data, the possibility to fetch all relevant attribute values is given to the user by querying AddressMaster.

Data that is connected in AM like Address, Coordinate, Property, Building. This possibility will enhance data quality if existing PointID could be found and less duplicate entries are created.

The following types of read operations towards AddressMaster will be requested:

* Read using a Point-Id
* Read using X and Y coordinates. The values are in SWEREF99 format and AddressMaster will do proper conversions and potential roundings.
* Read using a Property ID
* Read using any combination of attributes present in ENE

***Request 1***

Address information is read using a Point-Id. This is a request which quickly verifies whether a Point-Id already exists.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Mandatory | Type | Description | Example |
| reqHeader | yes | reqHeaderType | Message header | See data type section |
| pointId | yes | NUMBER | Point-Id of the requested address | 100003711 |

***Request 2***

Address information is read using X and Y coordinates.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Mandatory | Type | Description | Example |
| reqHeader | yes | reqHeaderType | Message header | See data type section |
| x | yes | VARCHAR2(20 BYTE) | SWEREF99 format of the X coordinate. Assuming AM does the coordinate conversion. | E18° 06 28.4400 E18°06.474 18.1079 |
| y | yes | VARCHAR2(20 BYTE) | SWEREF99 format of the Y coordinate. Assuming AM does the coordinate conversion. | N59°14 29.0399 N59°14.484 59.2414 |

***Request 3***

Address information is read using property ID.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Mandatory | Type | Description | Example |
| reqHeader | yes | reqHeaderType | Message header | See data type section |
| propertyId | yes | VARCHAR2(9 BYTE) | The identity of the requested property | 210067362 |

***Request 4***

Address information is read using any combination of attributes in eneAddressRecordType. For example, combining values for

* Street name
* Street number
* Postal code
* Postal area/city

will narrow the search as a logical “AND” is used with several search terms. A search is not limited to those attributes listed in the example above.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Mandatory | Type | Description | Example |
| reqHeader | yes | reqHeaderType | Message header | See data type section |
| addressRequestParams | yes | eneAddressRecordType | Attributes and attribute information that are sent to AM for an search operation. | See data type section |

***Response***

The response from AM/GESB is as many attributes that can be populated in the amFullAddressType. Even the parameters received as **input** will be used when sending the output information, meaning that some data might be transmitted both ways.

If a search should result in several addresses, an error message is returned if the number of results exceeds a predefined limit.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Mandatory | Type | Description | Example |
| resHeader | yes | resHeaderType | Response header, reporting same requestID of request, timestamp and other fields to manage request/response mechanism.  This entity will contain also response result and possible error codes and messages | See data type section |
| amAddressRecord | yes | amFullAddressParamsType | AddressMaster will return as much information as possible based on the request record. | See data type section |

### getAmBuilding

To get more detailed Building info.

***Request***

Get building information for a Point-Id.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Mandatory | Type | Description | Example |
| reqHeader | yes | reqHeaderType | Message header | See data type section |
| pointId | yes | NUMBER | Point-Id of the address to the building | 100003711 |

***Response***

The response is all building information for a Point-Id.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Mandatory | Type | Description | Example |
| resHeader | Yes | resHeaderType | Response header, reporting same requestID of request, timestamp and other fields to manage request/response mechanism.  This entity will contain also response result and possible error codes and messages | See data type section |
| amBuildingRecord | Yes | amBuildingType | Attributes related to building |  |
| amEntranceParams | Yes | amEntranceParamsType | Attributes related to all entrances for a specific building |  |

### getAmProperty

Get full property information for a single property, based on search with Point-ID or property coordinates.

***Request 1***

Get a property based on its Point-Id.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Mandatory | Type | Description | Example |
| reqHeader | yes | reqHeaderType | Message header | See data type section |
| pointId | yes | NUMBER | Point-Id of a property | 100003711 |

***Request 2***

The response is all property information for a Point-Id.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Mandatory | Type | Description | Example |
| reqHeader | yes | reqHeaderType | Message header | See data type section |
| x | yes | VARCHAR2(20 BYTE) | A coordinate in SWEREF99 format to be used as search criteria. | E18° 06 28.4400 E18°06.474 18.1079 |
| y | yes | VARCHAR2(20 BYTE) | A coordinate in SWEREF99 format to be used as search criteria. | N59°14 29.0399 N59°14.484 59.2414 |

***Response***

Respond with property information. In addition, also related buildings, their entrances, and polygon information is returned in the response.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Mandatory | Type | Description | Example |
| resHeader | Yes | resHeaderType | Response header, reporting same requestID of request, timestamp and other fields to manage request/response mechanism.  This entity will contain also response result and possible error codes and messages | See data type section |
| amPropertyParams | Yes | amPropertyType | Attributes related to the found property | See data type section |
| propertyArea | no | amPolygonCoordinateParamsType | The polygon information for a property | See data type section |
| amPropertyBuilding | yes | amBuildingParamsType | All related buildings for a property identified by property id | See data type section |
| amPropertyEntrance | yes | amEntranceParamsType | All entrances related to a property, with sorting order being the related building | See data type section |

### getAmWalkway

Get walkway path information for a Point-Id. The walkway path description and entrance code information is already stored in ENE, but additional information is requested from AM.

***Request***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Mandatory | Type | Description | Example |
| reqHeader | yes | reqHeaderType | Message header | See data type section |
| pointId | yes | NUMBER | Point-Id of a walkway path | 100003711 |

***Response***

The response is all walkway path for a Point-Id. As the walkway path information is associated to a chassis, it is treated with a service of its own.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Mandatory | Type | Description | Example |
| resHeader | Yes | resHeaderType | Response header, reporting same requestID of request, timestamp and other fields to manage request/response mechanism.  This entity will contain also response result and possible error codes and messages | See data type section |
| amWalkwayParam | Yes | amWalkwayType | Attributes for walkway information |  |

# REST Services ENE.GESB

The services in this section are those that are provided from ENE to GESB, i.e. ENE is the server (or provider) and GESB is the client (or consumer). ENE capabilities for creation and update are limited to those attributes present in ENE database.

## Create Services

### createAmAddress

Create a new Local Address or on Apartment Number level (could also be a SDU or Business CAP) for a Customer Site or Coordinate Address for a Network object if it is missing in AM.

Addresses of the category GA are not created using ENE.

Upon request of the creation of an address ENE will not have an assigned punkt\_id. As such, the ENE address will initially be created as a non-verified address. As mentioned in the details above, once an Address Master punkt\_id is assigned from Address Master, the GESB should utilize the ENE REST endpoint for updating an address and provide the new punkt\_id.

***Message***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Mandatory | Type | Description | Example |
| pubSubHeader | yes | publishSubscribeHeader | Message header | See data type section |
| addressRequestParams | yes | eneAddressRecordType | Based on what type of address is created, different subsets of attributes in this record are filled in. |  |

### createAmWalkwayPath

When a walkway path is needed in ENE-EGI it shall be able to create in AM to get PointID and to connect to Address Point ID or other Network Object without PointID. AM shall store info about the type of object it is (ex FP, SP).

***Message***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Mandatory | Type | Description | Example |
| pubSubHeader | yes | publishSubscribeHeader | Message header | See data type section |
| pointId | no | NUMBER | An existing Point-Id, or empty value if a new Point-Id is to be created. | 100003711 |
| walkwayParams | yes | amWalkwayType | The walkwayObjectType attribute needs to be aligned with AM. The typical usage is that an ENE user creates walkway paths. | See data type section |

## Update Services

### updateAmAddress

Update of a Local Address on Floor or Apartment Number level (could also be a SDU or Business CAP) for a Customer Site or Coordinate Address for a Network object that ENE-EGI is using.

***Message 1***

The update request is performed on a Point-Id, and the attributes to be updated, potentially a subset, are in the addressRequestParams parameter.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Mandatory | Type | Description | Example |
| pubSubHeader | yes | publishSubscribeHeader | Message header | See data type section |
| pointId | yes | NUMBER | Point-Id of the requested address | 100003711 |
| addressRequestParams | yes | eneAddressRecordType | Attributes and attribute information that are sent to AM for an update. |  |

***Message 2***

This request updates the X and Y for an address identified by a Point-Id. The precision of the attributes is chosen by the user.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Mandatory | Type | Description | Example |
| pubSubHeader | yes | publishSubscribeHeader | Message header | See data type section |
| pointId | yes | NUMBER | Point-Id of the requested address | 100003711 |
| x | yes | VARCHAR2(20 BYTE) | A new coordinate in SWEREF99 format to be added to the Point-Id. This will lead to that AM sets the Coordinate\_Type to V (Verified by ENE). |  |
| y | yes | VARCHAR2(20 BYTE) | A new coordinate in SWEREF99 format to be added to the Point-Id. This will lead to that AM sets the Coordinate\_Type to V (Verified by ENE). |  |

### updateAmWalkwayPath

Update of information related to a walkway path.

***Message***

The update request is performed on a Point-Id, and the attributes to be updated, potentially a subset, are in the addressRequestParams parameter.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Mandatory | Type | Description | Example |
| pubSubHeader | yes | publishSubscribeHeader | Message header | See data type section |
| pointId | yes | NUMBER | Point-Id of the walkway path to be updated | 100003711 |
| walkwayParams | yes | amWalkwayType | This structure contains the new parameters to be updated. | See data type section |

## Delete Services

### deleteAmAddress

Only delete the connection between PointID and ENE-EGI object of the subscription of the PointID. The message will not trigger an actual delete in AM.

***Message***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Mandatory | Type | Description | Example |
| pubSubHeader | yes | publishSubscribeHeader | Message header | See data type section |
| pointId | yes | NUMBER | Point-Id of the address to be deleted | 100003711 |

# SOAP/XML Services ENE.ENE

The services in this section are those that are provided from ENE to ENE, i.e. ENE is the server (or provider) and ENE is the client (or consumer). ENE capabilities for creation and update are limited to those attributes present in ENE database.

For the purpose of not duplicating information from the above sections, the datatypes for the SOAP services will be the same as the services defined in Section [4].

The sample messages in the Appendix will define the actual XML format used for these messages.

# AddressMaster Response Code Handling

The purpose is to describe the different response codes and their descriptions. These responses are received from AM, via GESB. Any responses are displayed in ENE based on the context of the action taken. In some cases this will result in a log entry, and in other cases there will be immediate feedback.

Only the Swedish descriptions are shown, even though AM has most of the error explanations in English also.

|  |  |
| --- | --- |
| RETURKOD | MEDDELANDE |
| 0 | OK |
| 10 | Punkt-id ej numeriskt |
| 11 | Anropande system ej behörigt |
| 12 | Punkt-id saknas i AM |
| 13 | City must be specified (max 50 characters) |
| 13 | Postort maste anges (Max 50 tkn) |
| 14 | Postort saknas i AM |
| 15 | Gatunamn saknas i AM |
| 16 | Gatunummer saknas i AM |
| 17 | Ingång saknas i AM |
| 18 | Uppgång saknas i AM (eller överliggande nivå i anropet) |
| 19 | Trappantal saknas i AM (eller överliggande nivå i anropet) |
| 20 | TV/TH saknas i AM (eller överliggande nivå i anropet) |
| 21 | Lägenhetsnummer saknas i AM (eller överliggande nivå i anropet) |
| 22 | Det finns fler än 50 adresser på närmast underliggande nivå |
| 23 | Ogiltigt gatunummer |
| 24 | Ogiltig ingång |
| 25 | Ogiltig uppgång |
| 26 | Ogiltigt trappantal |
| 27 | Ogiltigt TV/TH |
| 28 | Postnummer saknas i AM |
| 29 | Flera gator med samma namn finns inom angiven postort (postnummer måste anges) |
| 30 | Postnummer måste anges vid uppläggning av nytt gatunummer |
| 31 | Gatunamn måste anges (max 50 tkn) |
| 32 | Gatunummer måste anges |
| 33 | Ingång måste anges |
| 34 | Uppgång måste anges |
| 35 | Trappantal måste anges |
| 36 | TV/TH måste anges |
| 37 | Postnummer måste anges vid uppläggning av ny gata (en uppläggning per postnummer om gatan finns på flera postnummer) |
| 38 | Ogiltigt anropande system-id |
| 39 | Ogiltigt Postnummer |
| 40 | Ogiltigt lägenhetsnummer |
| 41 | Adressen finns redan registrerad i AM |
| 42 | Det finns fler än 100 adresser som motsvarar sökvillkoren |
| 43 | Ogiltig koordinattyp |
| 44 | Ogiltig landskod |
| 45 | Ogiltigt koordinatsystem |
| 46 | Orimligt värde på koordinater |
| 47 | Fält får ej vara tomt |
| 48 | Denna landskod finns redan |
| 49 | Denna returkod finns redan |
| 50 | Landskod saknas i AM |
| 51 | Nytt Alias är ej angivet |
| 52 | Nytt Alias är nu registrerat! |
| 53 | Alias är redan registrerat! |
| 54 | Ogiltigt antal adresser |
| 55 | Läns- och kommunkod saknas i AM |
| 56 | Församlingskod saknas i AM |
| 57 | Svaret innehåller för många träffar. Precisera sökvillkoren! |
| 58 | Ogiltig länskod |
| 59 | Ogiltig kommunkod |
| 60 | Ogiltig församlingskod |
| 61 | Länskod saknas i AM |
| 62 | Kommunkod saknas i AM |
| 63 | Församlingskod saknas i AM |
| 64 | Adressen uppdaterades i AM |
| 65 | Fastighet ofullständigt ifyllt |
| 66 | Fastighet saknas i AM |
| 66 | Fastigheten saknas i AM |
| 67 | Sökta uppgifter saknas i AM |
| 68 | Koordinater saknas i AM |
| 69 | Adress saknas på fastigheten |
| 70 | Kan inte hitta två närliggande gatunummer |
| 71 | Station saknas i AM |
| 72 | Ingen site finns för angivna sökvilkor |
| 73 | Siteadress måste ha värde Y eller N |
| 74 | Kommun måste ha ett värde |
| 75 | Trakt måste ha ett värde |
| 76 | Koordinatsystem skall vara=R (RT90 2,5 gon v) |
| 77 | Koordinattyp skall vara=M (manuellt inlagd) |
| 78 | Uppdateringen kunde inte genomföras |
| 79 | Flera gator med samma namn finns inom angiven postort och postnummer |
| 80 | Lghnr-alias kan ej registreras |
| 81 | Adress är ej på lägsta nivån |
| 82 | Nytt Lghnr kan bara anges i kombination med befintligt Lghnr. |
| 83 | Ogiltigt XML-dokument. |
| 84 | Huto saknas |
| 85 | Anropet till webservices har misslyckats |
| 86 | Både Objektnamn och Objekttyp måste anges |
| 87 | Ogiltig objekttyp |
| 88 | Det får inte förekomma dubbletter av Objektnamn av samma Objkettyp |
| 89 | Prisnivå för svartfiber saknas |
| 90 | TAD-normerat gatunamn får inte väljas som en dubblettsynonym |
| 91 | Dubblettsynonym är redan knuten till adressen |
| 92 | Det sökta punkt-id refererar till annan information |
| 93 | Det sökta punkt-id refererar till en borttagen adress |
| 94 | Transaktion-id måste anges |
| 95 | Svaret innehåller mer än 300 träffar |
| 96 | TAD-N flaggan kunde inte uppdateras |
| 97 | Det finns inga dubblettsynonymer till det givna gatunamnet i AM |
| 98 | Det givna Transid finns inte i AM |
| 99 | Funktion måste anges |
| 100 | Kalla\_id måste anges |
| 101 | Detaljadress till den givna adressen saknas i AM |
| 102 | Den givna adressen är inte en gårdsadress |
| 103 | Adressen ingår inte i ett FTTH område. |
| 104 | Det givna FTTH området saknas i AM. |
| 105 | UTO Uppdateringen kunde inte genomföras i AM |
| 106 | Fastighet saknas till den givna x och y koordinaten |
| 107 | Postaladress saknas till den givna x och koordinaten |
| 108 | Region till adress saknas |
| 109 | Fastighetstyp till adress saknas |
| 110 | Region och Fastighetstyp till adress saknas |
| 111 | Fas måste vara numeriskt |
| 112 | FTTH område måste anges |
| 113 | Lägenhetsnummer är redan registrerat på en annan trappa. |
| 114 | Det finns ingångar under detta gatnr, välj rätt ingång! |
| 115 | Lghnr är redan registrerat direkt under gatnr, kontrollera värde på Ingång! |
| 116 | Lghnr existerar redan under en ingång! |
| 117 | Det är inte tillåtet att registrera adressen på den nivå |
| 118 | Det givna FTTH området kan inte tas bort |
| 119 | Kommun till det givna FTTH området kan inte hittas i AM |
| 120 | Futo saknas |
| 122 | Lghnr från LMV/skatteverket får inte uppdateras! |
| 124 | Kandidatadress till det givna punkt-id saknas i AM |
| 125 | Adressen finns i AM, men den är ej en kandidatadress |
| 800 | Nätfråga kunde inte uppdateras |
| 998 | Ett tekniskt fel har inträffat |

# GESB Target responseCode Handling

Purpose of this section is to describe and decode “responseCode” field provided as output of each service in the EGI catalogue designed.

This section describes synchronous responses to a web service.

<resHeader> contains a further XML structure <responseResult> (type <responseResultType>) tag. This tag is composed by three fields for coding outcomes result and error handling. In particular responseCode is a char with 3 digit:

* =“000” in case outcome (i.e., responseOutcome>) is “OK”. This is the default value.
* <>“000” to report
  + Comments for <responseOutcome> = “OK”
  + Error codes in case <responseOutcome> = “KO”

In table below all decoded values for <responseCode>.

|  |  |  |  |
| --- | --- | --- | --- |
| **responseOutcome** | **responseCode** | **responseMessage** | **Comment** |
| OK | 000 | N/A |  |
| OK | 001 | Address updated in ENE |  |
| KO | 999 | Generic application error |  |
| KO | 997 | Method not in catalogue |  |
| KO | 996 | Service response timeout |  |
| KO | 995 | requestID already present |  |
| KO | 101 | Missing mandatory field “[field\_name]” |  |
| KO | 102 | Field wrongly formatted ([“field\_name”]) |  |
| KO | 103 | ENE database unavailable |  |
|  |  |  |  |
|  |  |  |  |

# REFERENCES

1. 1/063 53-HSM 901 3168, AddressMaster Attribute Specifications for ENE
2. 1/102 62-HSM 901 3168, Ericsson Integrated Inventory Address Management HLD
3. 1/155 19-HSM 901 3168, Interface Specification TAD-N Replacement - EGI

# APPENDIX A – AM-ENE Terminology Mapping

|  |  |
| --- | --- |
| ENE Attribute name | AM Attribute name |
| amAliasAddressName | ALIAS\_NAMN |
| amAliasAddressTypeId | ALIAS\_TYP\_ID |
| pointId | PUNKT\_ID |
| city | POSTORT |
| postalCode | POSTNR |
| streetName | GATUNAMN |
| streetNumber | GATNR |
| staircase | UPPGANG |
| floor | TRAPPANTAL |
| tltr | TVTH |
| apartmentNumber | LGHNR |
| typeId | TYP\_ID |
| typeIdName | TYP\_NAMN |
| addressOriginalSourceId | URSPRUNG\_KALLA\_ID |
| tadn | TADN |
| countryCode | LANDSKOD |
| countryCodeName | LAND |
| countyCode | LANSKOD |
| countyCodeName | LAN |
| municipalityCode | KOMMUNKOD |
| municipalityCodeName | KOMMUN |
| communityCode | FORSAMLINGSKOD |
| communityCodeName | FORSAMLING |
| coordinateSystemId | KOORDSYST\_ID |
| coordinateSystemIdName | KOORDSYST\_BESKRIVNING |
| coordinateTypeId | KOORDTYP\_ID |
| coordinateTypeIdName | KOORDTYP\_BESKRIVNING |
| x or x\_coord | X |
| y or y\_coord | Y |
| z | Z |
| propertyId | FNR |
| propertyMunicipality | F\_KOMMUN |
| propertyQuarter | F\_TRAKT |
| propertyBlock | F\_BLOCK |
| colonChr | F\_TKN |
| propertyUnit | F\_ENHET |
| areaNumber | OMRNR |
| propertyX | X |
| propertyY | Y |
| propertyAreaPolygon | LOCATION |
| errorMsgReturnCode | RETURNKOD |
| errorMsgText | MEDDELANDE |
| areaCode | RIKTNR |
| addressChangeFrom | PUNKT\_ID\_FEL |
| addressChangeTo | PUNKT\_ID\_OK |
| changedAddressDate | REG\_DATUM |
| siteObjectId | OBJEKT\_ID |
| siteObjectTypeId | OBJEKT\_TYP\_ID |
| siteObjectTypeIdName | OBJEKT\_TYP |
| walkwayPathText | VAG\_BESKRIVNING |
| walkwayObjectType | OBJEKT\_TYP |
| entranceCode | PORTKOD |
| walkwayRegistrationDate | REG\_DATUM |
| amAddressSourceId | KALLA\_ID |
| amAddressNameId | KALLA\_NAMN |
| amAddressNameIdDescription | KALLA\_BESKRIVNING |
| amAddressOriginatingSourceName | URSPRUNG\_KALLA\_ID |
| buildingTypeName | BYGGNAD\_TYP |
| habitationType | TYPBEBYGG |
| buildingUuid | UUID |
| buildingStatus | BYGGSTAT KOD |
| buildingStatusDescription | BESKRIVNING |
| entranceId | ENTRE\_ID |
| entranceType | ENTRETYP TYP\_ID |
| entranceTypeDescription | ENTRE\_TYP |
| entranceStatusCode | ENTRESTAT KOD |
| entranceStatusDescription | ENTRESTAT BESKRIVNING |
| elevatorStatusCode | HISS STATUS\_ID |
| elevatorStatusDescription | HISS STATUS |
| entranceUuid | BYGGNAD\_UUID |
| coordinateSystemIdName | KOORDSYST\_BESKRIVNING |
| coordinateTypeIdName | KOORDTYP\_BESKRIVNING |

# APPENDIX B – Sample Messages

The below sections provide example messages for the different actions listed in the sections above. These samples are provided as a blueprint and only describe some of the above interactions.

## Outgoing Messages – ENE as the client

These messages will be sent by ENE to the exposed REST endpoints in the GESB. For the purpose of documentation, these messages are provided here. However, the expectation is that the definition of these endpoints is provided by the GESB and for further details reference the document Addressmaster REST API 2.5.docx.

### Create Example

POST service

|  |
| --- |
| URL |
| http://server:port/AM-API/addresses |
| Header |
| Accept: application/json |
| Body |
| {  ”postort”:”Ankeborg”,  ”postnr”:”12345”,  ”gatunamn”:”Von Ankas Gata”,  ”gatnr”:”2”,  ”lghnr”:”123”,  ”kalla\_id”:”10”,  ”statusLang”:”swe”  } |

Response

|  |
| --- |
| HTTP Header |
| Status Code: 201 Created Content-Length: 96 Date: Sat, 07 Nov 2015 07:49:08 GMT Server: Apache-Coyote/1.1  Location: http://server:port/AM-API/addresses/145373291 |
| Body |
| {  "punkt\_id":145373291,  "postort":"Ankeborg",  "postnr":”12345”,  "gatunamn":"Von Ankas Gata",  "gatnr":"2",  "ingang":"",  "uppgang":"",  "trappantal":"",  "lghnr":"123"  } |

### Search Example

GET service

|  |
| --- |
| URL |
| http://server:port/AM-API/addresses/138501849?detailLevel=2 |

Response

|  |
| --- |
| HTTP Header |
| Status Code: 200 OK  Content-Length: 250  Date: Sat, 07 Nov 2015 07:49:08 GMT  Server: Apache-Coyote/1.1 |
| Body |
| {  "punkt\_id":138501849,  "postort":"Ankeborg",  "postnr":"22222",  "gatunamn":"Kajsagatan",  "gatnr":"1",  "ingang":"A",  "uppgang":"",  "trappantal":"",  "lghnr":""  } |

GET service

|  |
| --- |
| URL |
| http://server:port/AM-API/addresses?postort=ankeborg&postnr=22222&gatunamn=kajsagatan&gatnr=1&detailLevel=2&statusLang=swe |

Response

|  |
| --- |
| HTTP Header |
| Status Code: 200 OK  Content-Length: 250  Date: Sat, 07 Nov 2015 07:49:08 GMT  Server: Apache-Coyote/1.1 |
| Body |
| [  {  "punkt\_id":138501849,  "postort":"Ankeborg",  "postnr":"22222",  "gatunamn":"Kajsagatan",  "gatnr":"1",  "ingang":"A",  "uppgang":"",  "trappantal":"",  "lghnr":""  },  {  "punkt\_id":138501850,  "postort":"Ankeborg",  "postnr":"22222",  "gatunamn":"Kajsagatan",  "gatnr":"1",  "ingang":"B",  "uppgang":"",  "trappantal":"",  "lghnr":""  }  ] |

### Update Example

PUT service

|  |
| --- |
| URL |
| http://server:port/AM-API/addresses/123456 |
| Header |
| Accept: application/json |
| Body |
| {  “gatunamn”:”Kajsasgata”,  “gatnr”:”7”  } |

Response

|  |
| --- |
| HTTP Header |
| Status Code: 200 OK  Content-Length: 250  Date: Mon, 09 Nov 2015 10:19:44 GMT  Server: Apache-Coyote/1.1 |
| Body |
| {  "punkt\_id":123456  } |

### Delete Example

DELETE service

|  |
| --- |
| URL |
| http://server:port/AM-API/addresses/138501849 |
| Header |
| Accept: application/json |

Response

|  |
| --- |
| HTTP Header |
| Status Code: 204 No Content  Content-Length: 250  Date: Mon, 09 Nov 2015 10:19:44 GMT  Server: Apache-Coyote/1.1 |

## Incoming Messages – ENE as the server

These messages will be received by ENE from the exposed REST endpoints and called by the GESB. As indicated above, the actions of both ENE and the GESB represent both client and server. For consistency, the message formats between the ENE endpoints and the GESB endpoints are being kept as closely aligned as possible. Please refer to the above section for the current formats.

## Internal Messages – ENE as the client and server

These messages will be used by the ENE REST endpoints to communicate any edit changes in the underlying database. For the purposes of clarity, the same attribute names are currently being used as defined by the REST endpoints. It may be necessary during implementation to alter these to be the ENE database column names. This decision will be made at design time and the samples updated when necessary.

### Search Example

<?xml version="1.0" encoding="UTF-8"?>

<!--Sample XML file for search of address information -->

<addressSearch-req request-id="987654321" to-address="NE" from-address="ENE" xmlns="http://www.telcordia.com/neia\_integration" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.telcordia.com/neia\_integration C:\AddressMasterInterface.xsd"> <enquiry-ne>

<addressSearchQueryPayload>

<address>

<!-- with searchType = LIKE, this will return all addresses in Address Master that have " Ankeborg " in their address. i.e. %Main% -->

<punkt\_id>138501849</punkt\_id>

<postort>Ankeborg</postort>

<postnr>22222</postnr>

<gatunamn>Kajsagatan</gatunamn>

</address>

<!—Additional parameters can be specified for case sensitive searches -->

<searchType>EXACT</searchType>

<caseSensitive>true</caseSensitive>

</ addressSearchQueryPayload >

</enquiry-ne>

</ addressSearch-req>

### Create Example

<?xml version="1.0" encoding="UTF-8"?>

<!--Sample XML file for create of address information -->

<addressCreate-req request-id="987654321" to-address="NE" from-address="ENE" xmlns="http://www.telcordia.com/neia\_integration" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.telcordia.com/neia\_integration C:\AddressMasterInterface.xsd"> <enquiry-ne>

<addressCreateQueryPayload>

<address>

<!—depending on the source of the create we may not have punkt\_id, if we don’t this address gets created non-verified-->

<punkt\_id>138501849</punkt\_id>

<postort>Ankeborg</postort>

<postnr>22222</postnr>

<gatunamn>Kajsagatan</gatunamn>

</address>

</ addressCreateQueryPayload >

</enquiry-ne>

</ addressCreate -req>

### Update Example

<?xml version="1.0" encoding="UTF-8"?>

<!--Sample XML file for update of address information -->

<addressUpdate-req request-id="987654321" to-address="NE" from-address="ENE" xmlns="http://www.telcordia.com/neia\_integration" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.telcordia.com/neia\_integration C:\AddressMasterInterface.xsd"> <enquiry-ne>

<addressUpdateQueryPayload>

<address>

<!—punkt\_id is required and if not present will result in an error-->

<punkt\_id>138501849</punkt\_id>

<postort>Ankeborg</postort>

<postnr>22222</postnr>

<gatunamn>Kajsagatan</gatunamn>

</address>

</ addressUpdateQueryPayload >

</enquiry-ne>

</ addressUpdate -req>

### Delete Example

<?xml version="1.0" encoding="UTF-8"?>

<!--Sample XML file for delete of address information -->

<addressDelete-req request-id="987654321" to-address="NE" from-address="ENE" xmlns="http://www.telcordia.com/neia\_integration" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.telcordia.com/neia\_integration C:\AddressMasterInterface.xsd"> <enquiry-ne>

<addressDeleteQueryPayload>

<address>

<!—punkt\_id is required and if not present will result in an error-->

<punkt\_id>138501849</punkt\_id>

</address>

</ addressDeleteQueryPayload >

</enquiry-ne>

</ addressDelete -req>