Technical Blueprint
Dataset GPC Horticulture 2.0

Life

Domain

Kingdom

Phylum

Class

Order

Family

Genus

Species
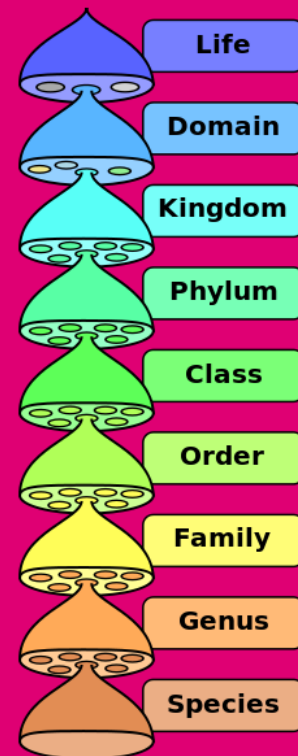
Functional description

How to implement the GPC Horticulture?

Content

Revision page

Document Version / Release number	Date	Review	Changes
Concept version 0.1	June 2015	Ben Ensink, Author	First Draft
Concept version 0.2	September 2015	GS1 Netherlands	Text corrections resulting from the GS1 Review
Concept version 0.3	September 2015	Floricode	Text corrections resulting from Floricode Review
Version 1.0	January 2016	Floricode & Ensink	Lookup table for the VBN-Group codes excluded; table PRODUCT_GPC included
Version 1.1	February 2016	Floricode	Corrections of the language codes
Version 1.2	March 2016	Floricode	Correction of the name of the Zip file Correction in table 4.12.3
Version 1.3	August 2016	Floricode	Addition for the FTP process on page 32
Version 1.4	February 2017	Floricode	Change Code list names in § 5.4 to the actual situation
Version 1.5	24-06-2021	Floricode	5.2.3. : change of the distribution channels
1.0/2.0	27-05-2024	Floricode	New release of this document based on the API data distribution of this Dataset GPC Horticulture

	Date	Name/organization	Initials
Approved by	27-05-2024	H. Zwinkels (Floricode) (Author)	HZ
Reviewed by	23-05-2024	M. Snippe (Floricode)	MS
	23-05-2024	E. Bakker (Floricode)	EB
	04-06-2024	F. Hogema (GS1)	FH

1.   Management Summary

This document is intended to guide you in implementing the Global Product Classification (GPC) reference data for the Horticultural industry. Much like the Linnaeus Technical Blueprint and all the code lists that Floricode distributes for our sector, the GPC set conforms to the general rules and processes of all other distribution sets of Floricode.
The GPC dataset itself is developed and maintained by GS1 under the Global Standards Management Process (GSMP) and published on their global website. GS1 publishes two versions per year of GPC, together with a Delta Sheet showing the changes per version. The publication format is different from the Floricode format.
Furthermore the GS1 publication has no link to the 'Linnaeus-world'.

For bringing together both worlds it was decided to distribute the GPC reference data in the formats of Floricode as is customary in our industry and include in the distribution set a file which links the information between Linnaeus codes and GPC.

Both Floricode and GS1 have reviewed and validated this document and the corresponding distribution of this dataset by the Floricode API services.
If you feel something is missing or needs correction, or if you have any questions regarding this material, please feel free to contact Floricode, who will be the Local Standards Manager for this document and the accompanying files.

2.   Introduction

2.1   General

The **GS1 Global Product Classification** (GPC) for the horticultural industry is based on the botanical structure of horticultural products (just like the Linnaeus system for most VBN codes). Botanical taxonomy is the most common and widely accepted way to classify living organisms. Botanists around the world already agree on the division in Genus and Species, based on the work of the Swedish botanist Carolus Linnaeus. This is supported by modern DNA research. Taxonomy is also used in commercial processes and therefore an important addition to the GS1 GPC standards.

For this reason GPC for the horticultural industry is built around the concept of genus and species. For every application all genus-species combinations can be linked to exactly one GPC brick code.

At the right side in *Figure 1* you can find all the levels within the **general GPC structure**. The first three levels (segment, family and class) will guide you to find a brick for your product. The brick is the central concept of the GS1 classification system.

You can learn more about GPC in general at the global website of GS1 in a document called "GPC, Development & Implementation".[1]

Figure 1 : General GPC Structure

The **GS1 Global Product Classification** (GPC) is a system that gives both sides of trading partner relationship a common language for grouping products in the same way. It ensures that products are classified correctly and uniformly, everywhere in the world. The business objectives of GPC are to:

* Support buying programs by allowing buyers to pre-select groups of applicable products
* Provide a common language for category management, thus speeding up reaction to consumer needs
* Be a key enabler of the Global Data Synchronisation Network
* Be a pivotal classification system between the information exchange parties

GPC is a rules-based, four-tier classification system for grouping products. The four tiers are Segment, Family, Class, and Brick (with Attribute types and Attribute values). A Brick identifies a category incorporating products (Global Trade Item Numbers (GTINs)) that serve a common purpose, are of a similar form and material, and share the same set of category attributes.

---

[1] GS1 GPC guidance: : https://www.gs1.org/standards/gpc

Level	Length	Syntax	Example
Segment codes	8	Two digit code followed by six zeros	10000000
Family codes	8	Two digit code preceded by Segment code and followed by four zeros	10200000
Class codes	8	Two digit code preceded by Segment and Family codes and followed by two zeros	10203000
Brick codes	8	Starts with a "1"	10000123
Attribute Type codes	8	Starts with a "2"	20000123
Attribute Value codes	8	Starts with a "3"	30000123

Table 1 : GPC coding

2.2   GPC for Horticulture

You can see the **Horticulture GPC hierarchy** in *Figure 2.* As you may notice the first four levels are implemented in such a way that they represent, in a simple way, the Linnaeus data model as defined in the Linnaeus Technical Blueprint that can be found at the Floricode website.[2]

Please keep in mind the following facts:

- Since there are so many genus – species combinations, a lot of combinations that only have a limited worldwide turnover, are linked to so-called 'catch-all' GPC Bricks, much like the current group codes '…other.. ' of the VBN. So only the most important combinations with respect to their worldwide turnover have their own GPC codes.

- All GS1 brick codes within the GS1 classification methodology are unique and thus will define the likewise unique upper levels of that classification hierarchy, consisting of segment, family and class definitions. In other words if you know the brick code you will know the content of the higher levels.
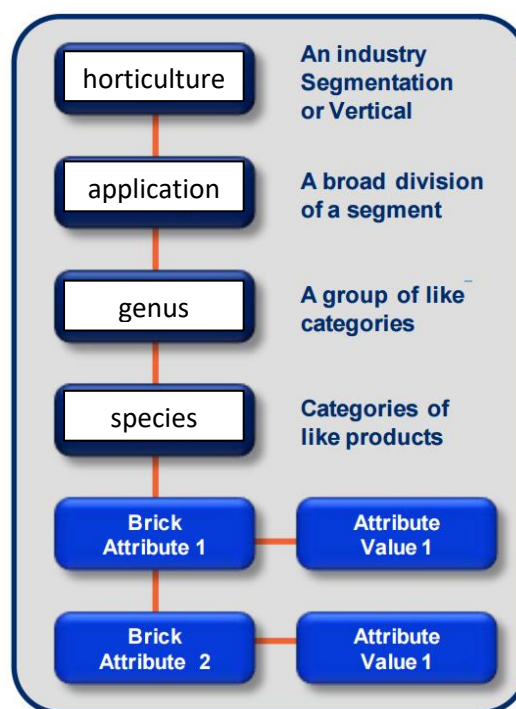
Figure 2 : GPC structure for Horticulture

- All 'living' or 'once living' products (*greenware*) are defined in the segment 'horticulture plants' of the GS1 methodology. All 'dead' accessories (*hardware*) like pots, vases etc. are defined in the segment 'Lawn/Garden Supplies', where a great number of tools and accessories were already defined. Of course products defined as 'dead' do not have a Genus and Species.

- When thinking about 'living' products, keep in mind that some product belong to a genus, which is not sub-divided into species. This is for instance the case with almost all product of the genus 'Rosa'.

---

[2] Linnaeus technical blueprint:
https://www.floricode.com/Portals/0/Downloads/STAMDATA/Productcodes/HZ17038V0_22%20FLC%20LINNAEUS%20model%20flowers%20and%20plants%202_20.pdf?ver=JIYiskc7B6mCPbXwDn9A7Q%3d%3d

- In *Figure 2.* you can see that the second level (under the segment horticulture) is called 'application' (Dutch; toepassing). This level is used to distinguish between cut flowers, cut greens and live plants, which closely resembles the normal way in our industry. Some products however do not belong to the definition of "cut-flowers, cut-greens or live plants" and therefore are classified in different GPC families (like seeds, bulbs etc.).

- The third and fourth levels, genus and species, should be familiar for all who work in our industry.

- The lower levels (Brick Attribute and Attribute value) may occur more than once and are comparable with the feature type codes and values (Dutch; kenmerken codes en waardes) known in the Linnaeus code system.

Complications encountered

A complication for finding a GPC of a product might be that in the registration database of Floricode, some products do not belong to one specific so-called 'Crop' (Dutch 'Gewas'). Look for instance at product_id 1713 'Ornamental grasses other', which obviously is not linked to any registration. This product belongs to group code 10789901 ('Ornamental grasses other') wherein several different Generae (plural for Genus) occur and therefore products in that group might belong to different GPC Bricks.

2.3   Linking VBN to GS1 GPC Horticulture

All horticulture products belong in the **Segment** "Horticulture Plants". No other products than these can be classified in this segment. See the Blue rectangle at the top in *Figure 3.*

GPC- **Family** ₎ the second (yellow) level in the GPC hierarchy, is equivalent to Application within VBN 'with a twist'. The application in VBN and the first position of the group code is 1 for cut flowers and cut greens, 2 for potted plants and 3 for garden plants. GPC however distinguishes between cut-flowers (1), cut greens (2) and live plants, containing both potted- and garden plants (3). Other GPC families exists for 'special' products like bulbs, trees, seeds etc.

The third level of the GPC classification hierarchy, called a **Class**, is that of the Genus (light blue). Because in the Horticultural industry everywhere in the world people are familiar with taxonomy and the Linnaean concept of Genus / Species, GS1 recognized the global use and understanding of this and built the GPC hierarchy around this concept.

The fourth level, also called **Brick** (Dutch: bouwsteen), finally arrives at the Genus-Species combinations referred to previously which is also familiar from the Linnaeus methodology. (green).

Any further characteristics of certain bricks can be defined by **Attribute type and value** pairs (not shown in *Figure 3*), like for instance 'Horticulture Color' and the colour value code for 'Pink'. These attribute-value pairs form the lowest, finest grained classification level of the GPC system. They are much like the feature type codes and values of the Linnaeus data model.
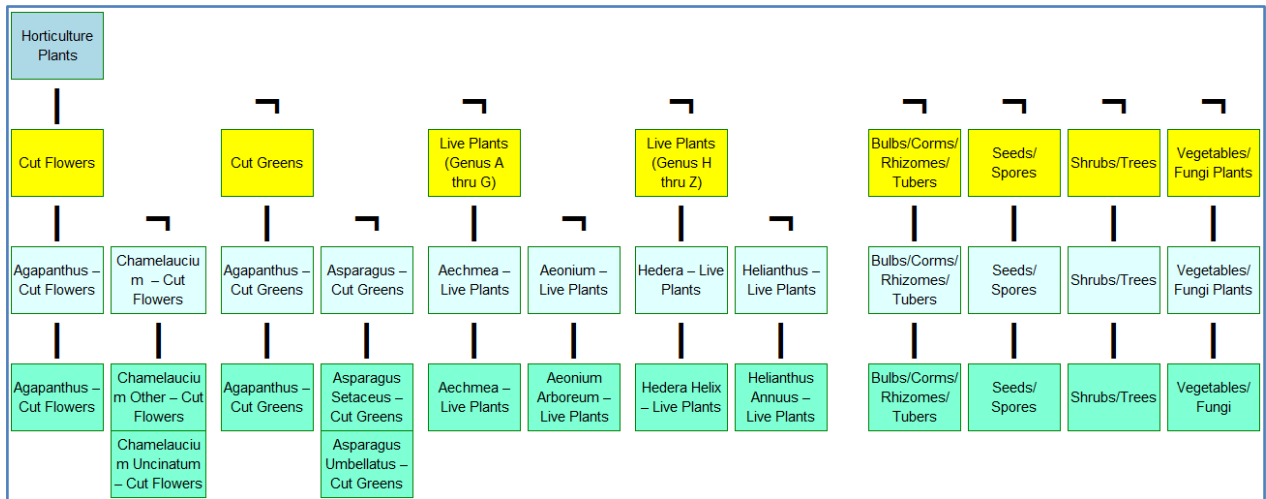
Figure 3 : GPC Horticulture : Visual map

3.   GPC Horticulture Data model

All code lists of the GPC Horticulture dataset (code lists) are displayed in *Figure 4* of the data model below. As you may notice the files 36 and 37[3] form the central part. Basically they define which upper three levels of the GPC code belong to which brick (36) and which attribute types and values may occur at what brick (37).

For a number of descriptions and definitions as well for the color-names, translations are available through the code lists NAME and NAME_TYPE. Since there are so many data fields translated any key or link was omitted here.

For the exact definition of keys and foreign keys please take a look at the corresponding entities descriptions in the next chapter.
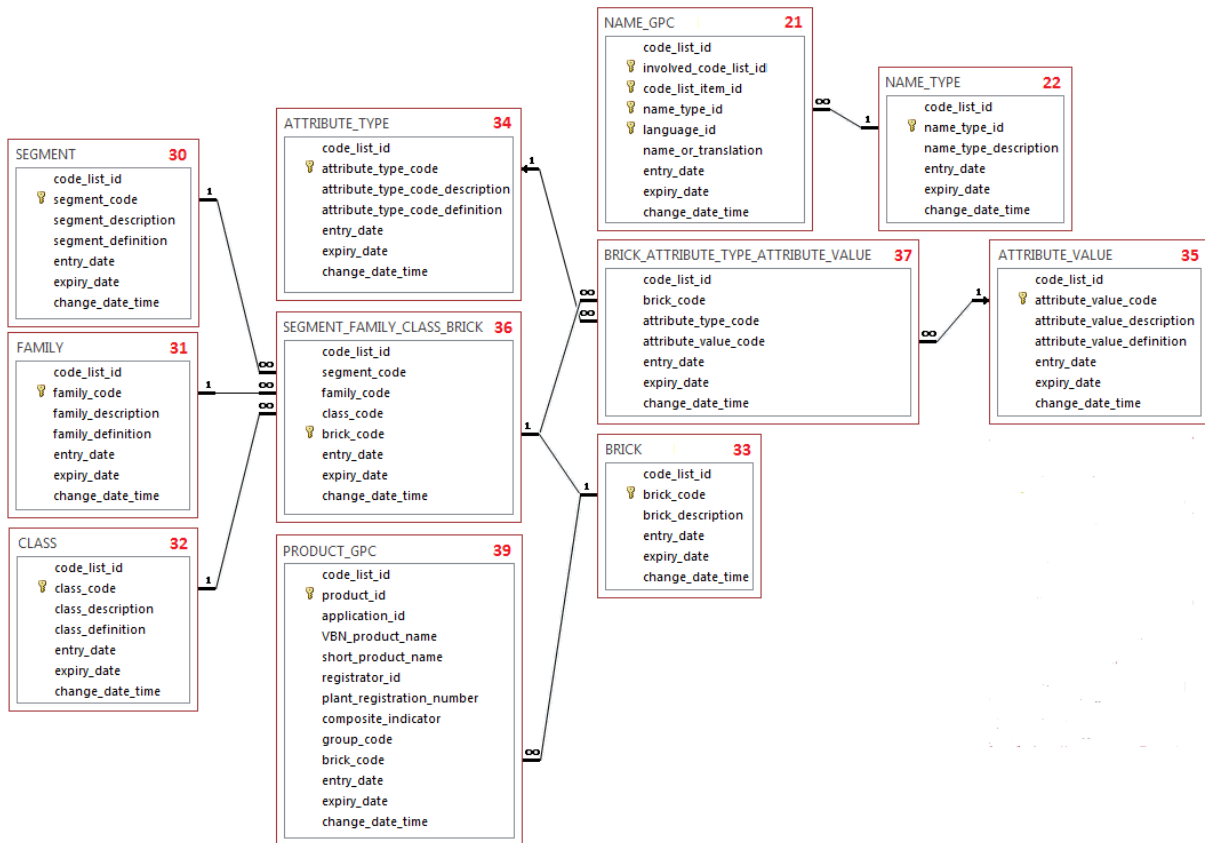
Figure 4 : Data model GPC Horticulture code lists

---

[3] The numbering of the datafiles cannot be found in the JSON API Response messages. Also the names of the attributes differ slightly from those in the JSON API response messages. This is documented in the JSON API documentation on the Floricode website.

4.   GPC Horticulture code list record layout

The GPC Horticulture code lists are available as API endpoints. This chapter describes the record layout of each file of the GPC Horticulture data model

4.1   Common code list aspects

All code lists exhibit the following generic characteristics.

4.1.1   Field names

It is assumed that the lists will also be applied by foreign business partners (growers, traders). The GPC cod list will therefore become available in various languages. The names of code lists and fields follow the ISO/IEC 11179 convention.

4.1.2   Update fields

Each item in any code list contains the following fields:

- entry date: the date at which a new item will or has become valid within the industry; this may be a future date.
- expiry date: the last effective date on which an item is still generally useful within the sector. This can indicate a future date or be left blank. An item is still considered "generally useful within the sector" on the expiration date itself.
- change date/time: the date and time at which an item was last updated (entered, changed or expired).

For further explanation about the use of these fields in detecting updates, please refer to 5.1.3 of this document.

4.1.3   Code list description

For each code list, the following information will be given:

- definition of the involved entity type
- description of the relationships with other entity types
- example of a filled-out code list item
- explanation or remark (if any).

For each code list entry the following will be specified:

- field number: field number, starting with 1
- field name: name of the field
- occurrence type: M = mandatory (field needs to be filled) or

    C = conditional (filled under certain conditions)

- format: N = numeric, followed by the number of digits

    AN = alphanumeric, followed by the number of characters

    ♦ fixed: N2   or   AN11

    ♦ variable: N..3   or   AN..35

- key fields: P#: primary key

    F#: foreign key

    PF#: both primary and foreign key

- content: additional remarks on that field pertaining to the content or format.

4.2   NAME_GPC

4.2.1   Definition

NAME_GPC is the translation code list for some description and definition fields. Currently the code list has translations and/or descriptions for:

Code list_id	Object of translation	Language(s)	code_list_item _id	Name_type _id
20	UPOV color description	Dutch, French, German, Spanish	10	1
30	Segment descriptions	Dutch	3	1
30	Family descriptions	Dutch	3	1
32	Class descriptions	Dutch	3	1
33	Brick Descriptions incl. common names Genus/Species	Dutch	3	1
33	Brick Definitions (include)	Dutch, English	3	5
33	Brick Definitions (exclude)	Dutch, English	3	6
34	Attribute Type descriptions	Dutch	3	1
35	Attribute Value descriptions	Dutch	3	1

Table 2 : Translations and descriptions available

With involved_code_list_id and involved_code_list_item_id a link to a code list and the specific field of the records within that code list can be indicated. Name_type_id indicates the purpose of a name (translation, description, …). Currently Dutch, French, German and Spanish translations for fields with 'descriptions' and Dutch translation for 'including and excluding definitions' are available.

4.2.2   Relationships

- NAME_GPC has a link to every code list that is included in the involved_code_list_id. The involved_code_list_item_id defines for which field in that code list a translation is available. language_id defines what language(s) is (are) available and name_type_id defines the purpose of the translated name within the code list.

In order to keep the data model as defined in *Figure 4 : Data model GPC Horticulture code* lists easily readable, these relationships have been left out of the data model.

4.2.3   Contents

The code list NAME_GPC has the following contents:

field	field_name	M/C	format	key	contents
1	code_list_id	M	N..3		'21'
2	involved_code_list_id	M	N..3	P#	
3	Involved_code_list_item_id	M	N..8	P#	
4	name_type_id	M	N..2	P#	
5	language_id	M	AN2	P#	
6	name_or_translation	M	AN..1024		
7	entry_date	M	N8		ccyymmdd
8	expiry_date	C	N8		ccyymmdd
9	change_date_time	M	N12		ccyymmddhhmm

Code list 21: NAME_GPC

4.2.4   Explanation

1:      code_list_id '21' is for NAME_GPC. The mean reason for this code list is to provide translations or added reference data.

2:      involved_code_list_id tells you for which code list a translation is available with this record.

3:      involved code_list_item_id tells you what the value of the primary key of the given code list is, for which a translation is available in this record. *(In the example the 23 indicates the UPOV_color_group_id 23, the item_id of the UPOV group where the translation of the UPOV_color_description belongs to)*

4:      name_type_id defines what the meaning of this translated text is.

5:      language_id defines the language of the translated text. Currently the following languages are presented

6:      name_or_translation holds the actual translated text for the referenced field in the language specified by the language_id

language_id	language name
DE	German (Duits)
FR	French (Frans)
NL	Dutch (Nederlands)
ES	Spanish (Spaans)
EN	English (Engels)

Table 3 : Translation Language id's and Names

4.2.5   Example

```
{
  "@odata.context": "string",
  "value": [
    {
      "involved_code_list_id": 20,
      "code_list_item_id": 23,
      "name_type_id": 1,
      "language_id": "ES",
      "name_or_translation": "rojo purpúreo",
      "entry_date": "1986-01-01",
      "change_date_time": "2016-03-03T00:00:00+01:00",
      "expiry_date": null
    }
  ]
}
```

4.3   NAME_TYPE

4.3.1   Definition

NAME_TYPE defines the purpose or target of a name in a certain language (translation). Currently name types are available for descriptions and definitions, though not in all code lists, since not every description or definition is currently available. See *Table 2 : Translations and* **descriptions available** for what is currently available.

4.3.2   Relationships

NAME_TYPE has one relation to NAME_GPC.

- One NAME-TYPE can be used in many NAME_GPC records.
- Many NAME_GPC records can point to one NAME_TYPE.

4.3.3   Contents

The code list NAME_TYPE has the following contents:

field	field_name	M/C	format	key	contents
1	code_list_id	M	N..3		'22'
2	name_type_id	M	N..2	P#	
3	name_type_description	M	AN..50		
4	entry_date	M	N8		ccyymmdd
5	expiry_date	C	N8		ccyymmdd
6	change_date_time	M	N12		ccyymmddhhmm

Code list 22 NAME_TYPE

4.3.4   Explanation

1:        code_list_id '22' is for NAME_TYPE.

2:        name_type_id defines where the translated text is applicable for. Currently that might be for a description, a definition include or a definition exclude expressed as 1, 5 or 6 respectively.

3:        name_type_description contains the meaning of the name type expressed in text.

Note: name types 2, 3 and 4 (Alternatieve naam, Afkorting; Korte Omschrijving) are not used for GPC translations. They are marked unused since they already exist in other datasets of Floricode.

4.3.5   Example

```
{
  "@odata.context": "string",
  "value": [
    {
      "id": 5,
      "description": "Definition (Include)",
      "entry_date": "2014-12-01",
      "change_date_time": "2014-12-01T12:00:00+01:00",
      "expiry_date": null
    }
  ]
}
```

4.4   SEGMENT

4.4.1   Definition

SEGMENT is the highest level in the classification hierarchy of GPC. It defines in what industry the products belong. For our industry the relevant segments are "Horticulture" and "Lawn/Garden supplies".

4.4.2   Relationships

SEGMENT has a link to the code list SEGMENT_FAMILY_CLASS_BRICK ("SFCB") which defines the hierarchy of the classification.

- One segment is linked to one or more records in the code list "SFCB".

4.4.3   Contents

The code list SEGMENT has the following contents:

field	field_name	M/C	format	key	contents
1	code_list_id	M	N..3		'30'
2	segment_code	M	N8	P#	
3	segment_description	M	AN..50		
4	segment_definition	M	AN..1024		
5	entry_date	M	N8		ccyymmdd
6	expiry_date	C	N8		ccyymmdd
7	change_date_time	M	N12		ccyymmddhhmm

Code list 30: SEGMENT

4.4.4   Explanation

1:        code_list_id '30' is for SEGMENT, the first or highest GPC hierarchy level.

2:        segment_code defines the GPC code for a segment within the GPC hierarchy., see for the coding principles *Table 1 : GPC coding.*

3:        segment_description contains the textual meaning of the segment.

4:        segment_definition contains the definition of this segment.

4.4.5   Example

```
{
  "@odata.context": "string",
  "value": [
    {
      "id": 93000000,
      "description": "Horticulture Plants",
      "definition": "Horticulture plants are used in plant cultivation for human
use. This segment includes non-food crops, such as flowers, trees and shrubs, and
grasses used in the floral industry, landscaping, and ornamental decoration. It
also includes growing plants used to produce food, such as, fruits, vegetables, and
mushrooms.",
      "entry_date": "2014-12-01",
      "change_date_time": "2014-12-01T12:00:00+01:00",
      "expiry_date": null
    }
  ]
}
```

4.5   FAMILY

4.5.1   Definition

FAMILY is the second highest level in the classification hierarchy of GPC. It defines a classification level which resembles, but is not quite the same as, the VBN application. See paragraph **2.3** for the right comparison between the VBN application (or first position of the VBN group code) and the GPC application.

4.5.2   Relationships

FAMILY has a link to the code list SEGMENT_FAMILY_CLASS_BRICK ("SFCB") which defines the hierarchy of the classification.

- One family is linked to one or more records in the code list "SFCB".

4.5.3   Contents

The code list FAMILY has the following contents:

field	field_name	M/C	format	key	contents
1	code_list_id	M	N..3		'31'
2	family_code	M	N8	P#	
3	family_description	M	AN..50		
4	family_definition	M	AN..1024		
5	entry_date	M	N8		ccyymmdd
6	expiry_date	C	N8		ccyymmdd
7	change_date_time	M	N12		ccyymmddhhmm

Code list 31: FAMILY

4.5.4   Explanation

1:       code list '31' is for FAMILY, the second highest GPC hierarchy level.

2:       family code defines the GPC code for a family within the GPC hierarchy and is used for GPC. See for the coding principles *Table 1 : GPC coding.*

3:       family description contains the textual meaning of the family.

4:       family definition contains the definition of this family.

family_code	family_description
81010000	Lawn/Garden Supplies
93010000	Cut Flowers
93020000	Cut Greens
93030000	Live Plants (Genus A thru G)
93040000	Live Plants (Genus H thru Z)
93050000	Bulbs/Corms/Rhizomes/Tubers
93060000	Plants Variety Packs
93070000	Seeds/Spores
93080000	Shrubs/Trees
93090000	Vegetables/Fungi Plants

Table 4 : All Horticulture GPC families

4.5.5 Example

```
{
  "@odata.context": "string",
  "value": [
    {
      "id": 93010000,
      "description": "Cut flowers",
      "definition": "Includes products that may be described/observed as flowers
cut from growing plants that are used as fresh or dried flowers. These flowers may
include some foliage from the same plant such as a stem and/or leaves. Excludes Cut
Greens and combinations from more than one GPC family.",
      "entry_date": "2014-12-01",
      "change_date_time": "2014-12-01T12:00:00+01:00",
      "expiry_date": null
    }
  ]
}
```

4.6   CLASS

4.6.1   Definition

CLASS is the third highest level in the classification hierarchy of GPC. It defines the Genus and Application of all products on this level.

4.6.2   Relationships

CLASS has a link to the code list SEGMENT_FAMILY_CLASS_BRICK ("SFCB") which defines the hierarchy of the classification.

- One class is linked to one or more records in the code list "SFCB".

4.6.3   Contents

The code list CLASS has the following contents:

field	field_name	M/C	format	key	contents
1	code_list_id	M	N..3		'32'
2	class_code	M	N8	P#	
3	class_description	M	AN..50		
4	class_definition	M	AN..1024		
5	entry_date	M	N8		ccyymmdd
6	expiry_date	C	N8		ccyymmdd
7	change_date_time	M	N12		ccyymmddhhmm

Code list 32: CLASS

4.6.4   Explanation

1:          code_list_id '31' is for CLASS, the third highest GPC hierarchy level.

2:          class_code defines the GPC code for a class within the GPC hierarchy.and is used for the Genus, see for the coding principles *Table 1 : GPC coding.*

3:          class_description contains the textual meaning of the class.

4:          class_definition contains the definition of this class.

Note: The Class definition also includes aliases or common names for the Genus, which might improve recognition of the Genus. However take care with this, since common names might be tricky and confusing. That was the whole idea of Linnaeus to use unique (Latin) names.

4.6.5   Example

```
{
  "@odata.context": "string",
  "value": [
    {
      "id": 93041300,
      "description": "Hydrangea - Live Plants",
      "definition": "Plants specifically of the genus Hydrangea: also known as
hydrangea or hortensia.",
      "entry_date": "2014-12-01",
      "change_date_time": "2014-12-01T12:00:00+01:00",
      "expiry_date": null
    }
  ]
}
```

4.7   BRICK

4.7.1   Definition

BRICK is the fourth highest level in the classification hierarchy of GPC. It defines the genus-species combination of all products on this level. Some botanical products may not have a species (the genus is not sub-divided into species because this division is not present or not relevant commercially). In those cases the Brick will only contains the Genus.

4.7.2   Relationships

BRICK has a link to the code list SEGMENT_FAMILY_CLASS_BRICK ("SFCB") which defines the upper hierarchy of the classification.

• One brick is linked to exactly one record in the code list "SFCB".

4.7.3   Contents

The code list BRICK has the following contents:

field	field_name	M/C	format	key	contents
1	code_list_id	M	N..3		'33'
2	brick_code	M	N8	P#	
3	brick_description	M	AN..50		
4	entry_date	M	N8		ccyymmdd
5	expiry_date	C	N8		ccyymmdd
6	change_date_time	M	N12		ccyymmddhhmm

Code list 33: BRICK

4.7.4   Explanation

1:      code_list_id '33' is for BRICK. A brick to many people is 'the' GPC of a product, mainly because all higher levels of the hierarchy can be determined without looking at the product itself and all lower levels of the hierarchy are optional. Brick codes are unique, regardless under which higher levels they belong.

2:      brick_code defines the GPC code for a brick within the GPC hierarchy, used for the Genus-Species combination, see for the coding principles *Table 1 : GPC coding.*

3:      brick_description contains the textual meaning of the brick.

Note 1: Brick definitions are listed in the NAME_GPC code list, because they consist of two separate definition fields, one for the Include and one for the Exclude definition. Translations of two types of definition field in one code list might be confusing. See *Table 2 : Translations and* **descriptions available** for the available fields for Bricks

Note 2: As with the Genus, for Genus/Species combinations also, **aliases or common names** are provided in the definition fields. The Dutch translation also includes common names in the definitions.

4.7.5 Example

```
{
  "@odata.context": "string",
  "value": [
    {
      "id": 10006648,
      "description": "Hydrangea Macrophylla - Live Plants",
      "entry_date": "2014-12-01",
      "change_date_time": "2014-12-01T12:00:00+01:00",
      "expiry_date": null
    }
  ]
}
```

4.8   SEGMENT_FAMILY_CLASS_BRICK

4.8.1   Definition

SEGMENT_FAMILY_CLASS_BRICK is a code list whose purpose it is to define the upper hierarchic level of the GPC code system. It defines which segments are linked to which families and for each family to which classes and for each class to which bricks it belongs. It is the 'flattened" definition of that hierarchy and contains one record for each complete branch of the hierarchy tree. Therefore the total number of Bricks in the Horticulture segment defines the number of records in this code list.

4.8.2   Relationships

SEGMENT_FAMILY_CLASS_BRICK ("SFCB") has links to all upper level hierarchical components of GPC.

- More records of "SFCB" are linked to one record in SEGMENT
- More records of "SFCB" are linked to one record in FAMILY
- More records of "SFCB" are linked to one record in CLASS
- One record in "SFCB" is linked to one record in BRICK
- One record in "SFCB" may be linked to 0 or more records in PRODUCT_GPC
- One record in "SFCB" is linked to one or more records in

BRICK_ATTRIBUTE_TYPE_ATTRIBUTE_VALUE ("BATAV")

4.8.3   Contents

The code list SEGMENT_FAMILY_CLASS_BRICK has the following contents:

field	field_name	M/C	format	key	contents
1	code_list_id	M	N..3		'36'
2	segment_code	M	N8	F#	
3	family_code	M	N8	F#	
4	class_code	M	N8	F#	
5	brick_code	M	N8	P#	
6	entry_date	M	N8		ccyymmdd
7	expiry_date	C	N8		ccyymmdd
8	change_date_time	M	N12		ccyymmddhhmm

Code list 36 : SEGMENT_FAMILY_CLASS_BRICK

4.8.4   Explanation

1:       code_list_id '36' is for SEGMENT_FAMILY_CLASS_BRICK.

2:       segment_code defines a link to SEGMENT

3:       family_code defines a link to FAMILY

4:       class_code defines a link to CLASS

5:       brick_code defines a link to BRICK

4.8.5 Example

```
{
  "@odata.context": "string",
  "value": [
    {
      "segment_id": 93000000,
      "family_id": 93050000,
      "class_id": 93050100,
      "brick_id": 10003290,
      "entry_date": "2014-12-01",
      "change_date_time": "2014-12-01T12:00:00+01:00",
      "expiry_date": null
    }
  ]
}
```

4.9 ATTRIBUTE_TYPE

4.9.1 Definition

ATTRIBUTE_TYPE is part of the optional lower level hierarchy, comparable with the Dutch VBN Characteristics_type or feature type (sorteringskenmerk in Dutch). It allows users to further define, detail and classify products belonging to one brick on some predefined other characteristics. It may be that some users do not see the necessity for a finer granularity than that of a brick. For other bricks there might not be attributes defined as yet.

4.9.2 Relationships

ATTRIBUTE_TYPE has a link to the code list BRICK_ATTRIBUTE_TYPE_ATTRIBUTE_VALUE ("BATAV") which defines the lower hierarchy of the classification.

• One ATTRIBUTE_TYPE record is linked one or more records in the code list "SFCB".

4.9.3 Contents

The code list ATTIRBUTE_TYPE has the following contents:

field	field_name	M/C	format	key	contents
1	code_list_id	M	N..3		'34'
2	attribute_type_code	M	N8	P#	
3	attribute_type_code_description	M	AN..50		
4	attribute_type_code_definition	M	AN..1024		
5	entry_date	M	N8		ccyymmdd
6	expiry_date	C	N8		ccyymmdd
7	change_date_time	M	N12		ccyymmddhhmm

Code list 34**: ATTRIBUTE_TYPE**

4.9.4 Explanation

1:          code_list_id '34' is for ATTRIBUTE_TYPE.

2:          attribute_type_code defines the GPC code for an attribute type within the GPC hierarchy, used for special characteristics, see for the coding principles *Table 1 : GPC coding.*

3:          attribute_type_code_description contains the textual meaning of this attribute type code.

4:          attribute_type_code_definition contains the definition of this attribute type code.

4.9.5   Example

```
{
  "@odata.context": "string",
  "value": [
    {
      "id": 20000430,
      "description": "Indoor/Outdoor Use",
      "definition": "Indicates, with reference to the product branding, labelling
or packaging, the descriptive term that is used by the product manufacturer to
identify if the product is used indoors or outdoors.",
      "entry_date": "2014-12-01",
      "change_date_time": "2014-12-01T12:00:00+01:00",
      "expiry_date": null
    }
  ]
}
```

4.10   ATTRIBUTE_VALUE

4.10.1   Definition

ATTRIBUTE_VALUE is part of the optional lower level hierarchy, comparable with the Dutch Linnaeus Characteristics Value (kenmerk waarde in Dutch). Every attribute type may have multiple values attached to it. (*This is registered in BRICK_ATTRIBUTE_TYPE_ATTRIBUTE_VALUE.)* Although, for instance, some Plants or Flowers may not be available in all colors, every color is included for every brick. The attributes have, what they call, "generic" pick lists.

4.10.2   Relationships

ATTRIBUTE_VALUE has a link to the code list BRICK_ATTRIBUTE_TYPE_ATTRIBUTE_VALUE ("BATAV") which defines the lower hierarchy of the classification.

- One ATTRIBUTE_VALUE record is linked one or more record in the code list "SFCB".

4.10.3   Contents

The code list ATTRIBUTE_VALUE has the following contents:

field	field_name	M/C	format	key	contents
1	code_list_id	M	N..3		'35'
2	attribute_value_code	M	N8	P#	
3	attribute_value_description	M	AN..50		
4	attribute_value_definition	M	AN..1024		
5	entry_date	M	N8		ccyymmdd
6	expiry_date	C	N8		ccyymmdd
7	change_date_time	M	N12		ccyymmddhhmm

Code list 35 : ATTRIBUTE_VALUE

4.10.4   Explanation

1:       code_list_id '35' is for ATTRIBUTE_VALUE.

2:       attribute_value_code defines the GPC code for an attribute value within the GPC hierarchy., see for the coding principles *Table 1 : GPC coding.*

3:       attribute_value_description contains the textual meaning of this attribute value code.

4:       attribute_value_definition contains the definition of this attribute value code.

4.10.5   Example

```
{
  "@odata.context": "string",
  "value": [
    {
      "id": 30016235,
      "description": "FLOWERING",
      "definition": "Plant products that derive their decorative purpose from their
beautiful coloured flowers.",
      "entry_date": "2014-12-01",
      "change_date_time": "2014-12-01T12:00:00+01:00",
      "expiry_date": null
    }
  ]
}
```

4.11   BRICK_ATTRIBUTE_TYPE_ATTRIBUTE_VALUE

4.11.1   Definition

BRICK_ATTRIBUTE_TYPE_ATTRIBUTE_VALUE is a code list whose purpose is to define the lower hierarchic level of the GPC code system (feature type and feature value or in Dutch "kenmerk en kenmerkwaarde). It defines what Brick is linked to which Attribute types and which values are allowed for the Brick - Attribute Type combination.

4.11.2   Relationships

BRICK_ATTRIBUTE_TYPE_ATTRIBUTE_VALUE ("BATAV") has links to all lower level hierarchical components of GPC and links the Upper GPC levels to the Lower GPC levels.

- One or more "BATAV" records are linked to one brick in "SFCB"
- One "BATAV" record may be linked to 0 or more ATTRIBUTE_TYPE records.
- One "BATAV" record may be linked to 0 or more ATTRIBUTE_VALUE records

Note1: If a "BATV" record is linked to an ATTRIBUTE_TYPE, it is at least linked to 1 ATTRIBUTE_VALUE records.

Note2: Even though some bricks have no attributes, there still is a record for them in BATAV. The attribute type and attribute value in "BATAV" are empty in those cases.

4.11.3   Contents

The code list BRICK_ATTRIBUTE_TYPE_ATTRIBUTE_VALUE has the following contents:

field	field_name	M/C	format	key	contents
1	code_list_id	M	N..3		'37'
2	brick_code	M	N8	P#	
3	attribute_type_code	C	N8	PF#	
4	attribute_value_code	C	N8	PF#	
5	entry_date	M	N8		ccyymmdd
6	expiry_date	C	N8		ccyymmdd
7	change_date_time	M	N12		ccyymmddhhmm

Code list 37 : BRICK_ATTRIBUTE_TYPE_ATTRIBUTE_VALUE

4.11.4   Explanation

1:       code_list_id '37' is for BRICK_ATTRIBUTE_TYPE_ATTRIBUTE_VALUE.

2:	brick_code defines a link to BRICK

3:	attribute_type_code defines a link to ATTRIBUTE_TYPE

4:	attribute_value_code defines a link to ATTRIBUTE_VALUE

### 4.11.5	Example

```
{
  "@odata.context": "string",
  "value": [
    {
      "brick_id": 93000000,
      "attribute_type_id": 93050000,
      "attribute_value_id": 93050100,
      "entry_date": "2014-12-01",
      "change_date_time": "2014-12-01T12:00:00+01:00",
      "expiry_date": null
    }
  ]
}
```

## 4.12	PRODUCT_GPC

### 4.12.1	Definition

PRODUCT_GPC is a code list that provides a link between the Linnaeus world of product codes and that of the GPC coding system. The Linnaeus product code from Floricode (formally called VBN code) is well known and broadly in use in the floricultural sector. With this table the user can find the correct GPC code for each individual floricultural product which has a Floricode product code.

### 4.12.2	Relationships

PRODUCT_GPC has a link to the code list SEGMENT_FAMILY_CLASS_BRICK which defines the higher hierarchy of the classification.

- One PRODUCT_GPC record is linked to only one records in "BRICK".
- One "BRICK" record is linked to 0 or more records in PRODUCT_GPC

Note: some Brick codes are not related to any VBN Product, like the bricks for bouquets.

### 4.12.3	Contents

The code list PRODUCT_GPC has the following contents:

Field	Field_name	M/C	Format	Key	contents
1	code_list_id	M	N..3		'39'
2	product_id	M	N..7	P#	*PRODUCT* indentifier
3	application_id	M	N..2	F#	*APPLICATION* identifier
4	VBN_product_name	M	AN..105		Product name
5	short_product_name	M	AN..20		Short product name
6	registrator_id	C	N..2	F#	*PLANT* registrator id
7	plant_registration_number	C	N..7	F#	*PLANT* registration nr.
8	composite_indicator	M	N1		'0'
9	VBN_group_code	M	N8		VBN Group indentifier
10	brick_code	M	N8	F#	Brick code
11	entry_date	M	N8		ccyymmdd
12	expire_date	C	N8		ccyymmdd
13	change_date_time	M	N12		ccyymmddhhmm

Code list 39 : PRODUCT_GPC

4.12.4   Explanation

1:        code_list_id '39' is for PRODUCT.

2:        product_id is the Floricode (VBN) product code

3:        application_id defines under which application (cut flower, pot plant, garden plant) this product belongs; this refers to the code list *APPLICATION* in the distribution set of the VBN product codes

4:        VBN product name: if the product is a plant, as a rule, its name is made up of the concatenation of its genus name, species name (if any) and its cultivar name (if any); in some cases the name may consist of non-botanical elements

5:        display abbreviation: short name for (auction clock) display purposes

6:        ID of the organisation that has registered the plant, for example:
- 1: Floricode
- 2: Naktuinbouw

Together with field 7, this field constitutes a reference to code list *PLANT* in the distribution set of the VBN product codes

7:        taxonomic number under which the plant is registered by the registration authority

8:        indicator telling whether the product is an individual product or a mixed plant; meant to support possible future rules and constraints regarding mixed products, for example:
- 1:   individual product
- 2:   mixed plant

For the meanwhile this indicator is filled with '0'

9:        existing VBN group code.

10:        brick_code defines a link to BRICK, thus forming the link between VBN and GPC code systems.

4.12.5   Example

```
{
  "@odata.context": "string",
  "value": [
    {
      "brick_id": 10006648,
      "product_id": 655,
      "entry_date": "2006-01-12",
      "change_date_time": "2016-09-26T14:13:00+01:00",
      "expiry_date": null
    }
  ]
}
```

Note:
In the API response message on this code list only the brick_id and the product_id are mentioned. The other data mentioned in this code list (grey shaded) can be found at another API endpoint: /VBN/Product:

```
{
    "@odata.context": "string",
    "value": [    {
        "expiry_date": null,
        "entry_date": "1992-03-02",
        "id": 655,
        "application_id": 2,
        "name": "Hydrangea macrophylla 'Blauer Zwerg'",
        "short_name": "HYDR M BLAUER ZWERG",
        "plant_registrator_id": 1,
        "plant_taxonomic_number": 106577,
        "composite_indicator": 0,
        "product_group_id": 20203601,
        "change_date_time": "2017-04-26T16:39:00+02:00"
    }]
}
```

5.   Distribution of the dataset GPC Horticulture

This section will discuss policy matters and IT aspects of the distribution of the code lists of the dataset GPC Horticulture.

5.1   Distribution policy

With regards to the GPC distribution policy, the following principles apply.

5.1.1   Responsibilities

The contents of the dataset GPC Horticulture code lists is maintained by the agencies who are responsible for the GPC data management. These organisations are responsible for the correctness, completeness and the mutual consistency of the data. Floricode is specifically responsible for the Product-GPC file.

The code lists will be distributed by Floricode, who will only partly be responsible for the code lists' contents, namely the Product_GPC file. This implies that Floricode won't perform any further intrinsic processing on the data. Floricode will provide the technical platform and protocols to obtain the code lists. In practice, the distribution agent's job will be limited to:

- system management and problem solving by distributing the code lists
- ensuring the availability of the distribution platform
- registering users and distributing login-ID's and passwords (if applicable)
- providing support and technical information about the code lists.

5.1.2   Only current data

- From the starting point on of the distribution of this dataset with GPC code lists (January 2016) only the current actual data will be distributed. The main reason for this is the fact that the product codes are distributed completely (with the Current and Full files) in another data set of Floricode; in that data set the actual and historical truth of all items is secured.
- GS1 has the policy for GPC codes never to re-use any code previously used and having become obsolete.

5.1.3   Update detection

Another principle is that new, expired and changed items are to be detected easily within the complete dataset, without the involved business partners being bound to the update frequency of the distribution agent. The code list update fields will allow for this (see *4.1.2 Update fields*).

To establish which records are new, expired or changed, the code list processing application needs to record the last processing date and decide which items have been changed since that date, following the next procedure:

- new: change_date_time is later than the last processing date/time and entry_date is later than or equal to the change_date_time
- expired: change_date_time is later than the last processing date/time and expiry_date is later than or equal to the change_date_time
- changed: change_date_time is later than the last processing date/time and entry_date is before the change_date_time.

To warrant a proper functioning of this procedure, GS1 and Floricode will ensure that:

- new or expired items will not be entered or become expired retroactively, but only on the change date or later
- changes will not be made retroactively nor announced in advance, but effected on the change date only.
- once expired, items will not be reused.

5.1.4   Definition of new, expired and changed items

To be able to properly process the code lists, it's vital to strictly define the terms 'new', 'expired' and 'changed' items:

- new: an item, the ID of which has not yet been used before
- expired: an item with an existing ID, the expiry date of which has passed
- changed: an item with an existing ID, of which the contents of one or more of its attributes has been changed *in the database of the code list responsible agency*; since not all item attributes in the code list manager's system need to have been included in the distributed code lists, an item may be marked as changed without this being evident from any visible change in the fields of the involved code list item.

These definitions imply that items whose key fields have changed will be notified as being 'expired' (the replaced item) as well as being 'new' (the replacing item). If the change does not involve a key field but an attribute field, the item will go notified as 'changed'. The termination of items does not automatically mean that related items have also reached a deadline.

5.2   Distribution technique

With respect to the code list distribution technique, the following principles apply:

5.2.1   API response format

The GPC Horticulture code lists are available via the following API endpoints:

API Endpoint	Code_list_id
/FLC/GpcName	21
/FLC/GpcNameType	22
/FLC/GpcSegment	30
/FLC/GpcFamily	31
/FLC/GpcClass	32
/FLC/GpcBrick	33
/FLC/GpcAttributeType	34
/FLC/GpcAttributeValue	35
/FLC/GpcHierarchy	36
/FLC/GpcBrickAttribute	37
/FLC/GpcBricktoProduct	39

The API response works as follows:
After a API request the API responds with a message in JSON format.
Within a JSON message the records of a code list are an object within a range that starts with the name 'value'. Every object starts with '{' and ends with '}' . A range starts with '{' and ends with '}'. An object is a disordered set of name/value pairs. Each name is followed by ':' and the name/value pairs are separated by a ','.

Here is an example of an API response message:

```
{
    "@odata.context": "https://api.floricode.com/v2/flc/$metadata#GpcBrick",
    "value":    [
            {
        "expiry_date": null,
        "entry_date": "2014-12-01",
        "id": 10003290,
        "description": "Bulbs/Corms/Rhizomes/Tubers",
        "change_date_time": "2014-12-01T12:00:00+01:00"
    },
            {
        "expiry_date": null,
        "entry_date": "2014-12-01",
        "id": 10003291,
        "description": "Seeds/Spores",
        "change_date_time": "2014-12-01T12:00:00+01:00"
    }
    ]
}
```

Note that the code_list_id is not present In the API response message!

An API response message contains a maximum of 5000 objects (= records of a code list). To obtain more objects an iterative API call is necessary. One possibility to do that is the so called nextLink, the call by which the next set of a maximum of 5000 objects is retrieved. This nextLink is part of the API response message. Here is an example:

```
"@odata.nextLink": "https://api.floricode.com/v2/FLC/GpcBrickToProduct?$skip=5000"
```

It's possible to call for a limited number of objects by request for all the changes since a < date/time> or to ask for only the active objects.

5.2.2   Character set

Since the NAME_GPC code list will contain translations in languages that may use special (diacritic) characters, the Linnaeus code lists will support the UTF-8 character set (a subset of ISO/IEC 10646 Unicode). It will be up to those processing applications of the business partners, that do not support these characters, to replace special characters by appropriate dummies. In addition, the following rules/guidelines apply:

No semicolons in fields
To avoid possible conflicts with field separators, no code list field may contain a semicolon; the code list agent will see to this when producing the lists.

Characters in EDI messages
For the Floricode EDI messages (type EDIFACT), the current, limited UNOA and UNOB character sets will remain in place. Since EDI messages usually contain just codes and no descriptions or text, this restriction does not entail any functional limitation.

5.2.3   Distribution channel

This distribution set can be obtained as follows::

- Via API : https://api.floricode.com/v2/#

To get access to the API distribution service a subscription is required. Therefore an application form is available on the website of Floricode https://www.floricode.com/en-us/floricode/application-forms

6.   Abbreviations

BATAV Brick_attribute_type_attribute_value code list

CR-LF   Carriage Return / Line Feed

CSV   Comma Separated Values

DNA   Deoxyribonucleic acid

EAN   European Article Numbering

EDI   Electronic Data Interchange

FTP   File Transfer Protocol

GDSN   Global Data Synchronisation Network

GPC   Global Product Classification

GS1   Global Standards 1, formerly known as EAN

GSMP   Global Standards Management Process

GTIN   Global Trade Item Number

HTML   Hypertext Markup Language

LINNAEUS   Code system of the Dutch Horticultural sector, named after the Swedish botanist Carolus Linnaeus, now maintained by **Floricode**

RGB   Red Green Blue, coloring system for displays / computers

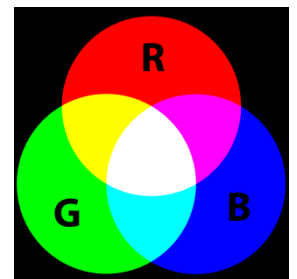RHS   the Royal Horticultural Society

SFCB   Segment_family_class_brick code list

UPOV   International Union for the Protection of New Varieties of Plants

VBN   Vereniging van Bloemenveilingen in Nederland

XLS   Microsofts Excel file format

XML   Extensible Markup Language

Annex 1: List of figures

Annex 2: List of tables