



Technical Blueprint

version 2.7

Cinnacus

version: 2.7
status: definitive
date: February 1, 2008
subject: Technical Blueprint of the renovated VBN Code system

© Vereniging van Bloemenveilingen in Nederland (VBN), 2008

All rights reserved. VBN rejects any liability for damages, directly or indirectly, of whatever nature, following from, or in any way related to applying the information contained in this document.

Summary

You are reading the 'Technical Blueprint' of the modernized VBN code system, the new method of identifying and specifying floricultural products and lots, adorned with the name 'Linnaeus' (the Latin name of the great taxonomist Carl von Linné).

This document is the technical translation and crystallization of an earlier (revised) 'conceptual' blueprint. That conceptual blueprint described in what way, and why, the new Linnaeus system differs from the current VBN code system. These changes pertain, in particular, to the:

- extension of the VBN product code from 5 to 7 positions
- addition and distribution of 'fixed' product features
- increase of the number of 'variable' lot features
- codification of rules and constraints governing lot specifications
- introduction of feature groups
- specification of composite lots
- support of multiple languages.

In practice, the new code system will be reflected in modifications of the external data exchange among supply chain business partners and of the internal business software used by the parties involved. The external data exchange adjustments involve both the structure, contents and processing of:

- fixed reference data captured in various code lists
- variable lot data exchanged on paper or through electronic messages

Depending on the various roles of the parties involved, their specific needs, demands and IT levels, business applications will need to be modified so as to:

- store and display the 7 digit product code
- check that lot specification rules and constraints are duly respected
- store and display multiple product- and lot features
- store and display parts of composite lots.

This technical blueprint will serve as a starting point for implementing these software changes. The document has been reviewed by representatives of all involved parties: VBN, auctions, growers and traders, and approved by the Linnaeus steering committee. It will form the basis for subsequent steps, such as:

- drafting new releases of the FlorEcom/Ediflower messages
- making a detailed Linnaeus migration plan
- setting up an environment for testing code lists and EDI messages
- integrating Linnaeus in the business applications of supply chain partners.

Changes

Relative to the previous version 2.6 this new version 2.7 has been changed on the following counts:

| Paragraph | section | change |
|------------------|----------------|---------------|
| 3.12.1 | | changed |
| 3.12.3 | | changed |
| 3.12.5 | | changed |
| 3.12.6 | | new |

TABEL OF CONTENTS

| | |
|--|-----------|
| SUMMARY | 3 |
| CHANGES | 4 |
| 1. INTRODUCTION | 8 |
| 1.1 BACKGROUND..... | 8 |
| 1.2 PURPOSE | 8 |
| 1.3 PRINCIPLES | 9 |
| 1.4 CONTENTS | 9 |
| 1.5 STATUS..... | 9 |
| 1.6 REFERENCES | 9 |
| 2. LINNAEUS DATA MODEL | 11 |
| 2.1 FIXED DATA | 11 |
| 2.2 VARIABLE DATA..... | 11 |
| 2.3 DATA MODEL NOTATION | 11 |
| 2.4 TECHNICAL DATA MODEL | 12 |
| 3. LINNAEUS CODE LISTS | 13 |
| 3.1 COMMON CODE LIST ASPECTS | 13 |
| 3.1.1 <i>Field names</i> | 13 |
| 3.1.2 <i>Update fields</i> | 13 |
| 3.1.3 <i>Code list description</i> | 13 |
| 3.2 PRODUCT..... | 14 |
| 3.2.1 <i>Definition</i> | 14 |
| 3.2.2 <i>Relationships</i> | 14 |
| 3.2.3 <i>Contents</i> | 14 |
| 3.2.4 <i>Example</i> | 14 |
| 3.2.5 <i>Explanation</i> | 14 |
| 3.3 APPLICATION | 15 |
| 3.3.1 <i>Definition</i> | 15 |
| 3.3.2 <i>Relationships</i> | 15 |
| 3.3.3 <i>Contents</i> | 15 |
| 3.3.4 <i>Example</i> | 15 |
| 3.3.5 <i>Explanation</i> | 15 |
| 3.4 PLANT | 15 |
| 3.4.1 <i>Definition</i> | 15 |
| 3.4.2 <i>Relationships</i> | 16 |
| 3.4.3 <i>Contents</i> | 16 |
| 3.4.4 <i>Example</i> | 16 |
| 3.4.5 <i>Explanation</i> | 16 |
| 3.5 GENUS | 16 |
| 3.5.1 <i>Definition</i> | 16 |
| 3.5.2 <i>Relationships</i> | 16 |
| 3.5.3 <i>Contents</i> | 16 |
| 3.5.4 <i>Example:</i> | 17 |
| 3.6 SPECIES | 17 |
| 3.6.1 <i>Definition</i> | 17 |
| 3.6.2 <i>Relationships</i> | 17 |
| 3.6.3 <i>Contents</i> | 17 |
| 3.6.4 <i>Example</i> | 17 |
| 3.7 CULTIVAR | 17 |
| 3.7.1 <i>Definition</i> | 17 |
| 3.7.2 <i>Relationship</i> | 17 |

| | | |
|--------|------------------------------|----|
| 3.7.3 | Contents | 17 |
| 3.7.4 | Example..... | 18 |
| 3.7.5 | Explanation | 18 |
| 3.8 | PRODUCT FEATURE | 18 |
| 3.8.1 | Definition | 18 |
| 3.8.2 | Relationships | 18 |
| 3.8.3 | Contents | 18 |
| 3.8.4 | Example..... | 18 |
| 3.8.5 | Explanation | 19 |
| 3.9 | FEATURE TYPE..... | 19 |
| 3.9.1 | Definition | 19 |
| 3.9.2 | Relationships | 19 |
| 3.9.3 | Contents | 19 |
| 3.9.4 | Example..... | 19 |
| 3.10 | FEATURE VALUE..... | 19 |
| 3.10.1 | Definition..... | 19 |
| 3.10.2 | Relationships | 19 |
| 3.10.3 | Contents | 20 |
| 3.10.4 | Example..... | 20 |
| 3.10.5 | Remark | 20 |
| 3.11 | FEATURE GROUP | 20 |
| 3.11.1 | Definition..... | 20 |
| 3.11.2 | Relationships | 20 |
| 3.11.3 | Contents | 20 |
| 3.11.4 | Example:..... | 20 |
| 3.12 | REGULATORY FEATURE TYPE..... | 21 |
| 3.12.1 | Definition..... | 21 |
| 3.12.2 | Relationships | 21 |
| 3.12.3 | Contents | 21 |
| 3.12.4 | Example..... | 21 |
| 3.12.5 | Explanation..... | 21 |
| 3.13 | REGULATION TYPE | 22 |
| 3.13.1 | Definition..... | 22 |
| 3.13.2 | Relationships | 22 |
| 3.13.3 | Contents | 22 |
| 3.13.4 | Example..... | 22 |
| 3.14 | NAME | 22 |
| 3.14.1 | Definition..... | 22 |
| 3.14.2 | Relationships | 22 |
| 3.14.3 | Contents | 23 |
| 3.14.4 | Example..... | 23 |
| 3.14.5 | Explanation..... | 23 |
| 3.15 | NAME TYPE..... | 24 |
| 3.15.1 | Definition..... | 24 |
| 3.15.2 | Relationships | 24 |
| 3.15.3 | Contents | 24 |
| 3.15.4 | Example..... | 24 |
| 3.15.5 | Explanation..... | 24 |
| 3.16 | LANGUAGE..... | 24 |
| 3.16.1 | Definition..... | 24 |
| 3.16.2 | Relationships | 24 |
| 3.16.3 | Contents | 24 |
| 3.16.4 | Example..... | 25 |
| 3.16.5 | Explanation..... | 25 |
| 3.17 | GROUP | 25 |
| 3.17.1 | Definition..... | 25 |
| 3.17.2 | Relationship..... | 25 |

| | | |
|-----------|---|-----------|
| 3.17.3 | Contents | 25 |
| 3.17.4 | Example | 25 |
| 3.17.5 | Explanation | 25 |
| 3.18 | PHOTO REFERENCE | 25 |
| 3.18.1 | Definition | 25 |
| 3.18.2 | Relationships | 25 |
| 3.18.3 | Contents | 26 |
| 3.19 | PHOTO TYPE | 26 |
| 3.19.1 | Definition | 26 |
| 3.19.2 | Relationships | 26 |
| 3.19.3 | Contents | 26 |
| 4. | CODE LIST DISTRIBUTION | 27 |
| 4.1 | DISTRIBUTION POLICY | 27 |
| 4.1.1 | Responsibilities | 27 |
| 4.1.2 | Complete set | 27 |
| 4.1.3 | Update detection | 28 |
| 4.1.4 | Special cases in the detection of updates | 28 |
| 4.1.5 | Definition of new, expired and changed items | 29 |
| 4.2 | DISTRIBUTION TECHNIQUE | 29 |
| 4.2.1 | File format | 29 |
| 4.2.2 | Character set | 30 |
| 4.2.3 | Distribution channel | 30 |
| 5. | LINNAEUS IN EDI MESSAGES | 31 |
| 5.1 | MAPPING THE LINNAEUS TECHNICAL DATA MODEL ON EDI MESSAGES | 31 |
| 5.1.1 | Lot | 31 |
| 5.1.2 | Base product | 31 |
| 5.1.3 | Part | 31 |
| 5.1.4 | Lot feature | 32 |
| 5.1.5 | Photo reference | 32 |
| 5.2 | EXTENSION OF LOT FEATURES | 32 |
| 5.3 | HARMONISATION OF THE EDI MESSAGES | 33 |
| 5.3.1 | Uniform specification of lot features | 33 |
| 5.3.2 | Uniform field formats | 33 |
| 5.3.3 | Extension of the VBN product code | 33 |
| 5.4 | SPECIFICATION OF COMPOSITE LOTS | 34 |
| 5.4.1 | Definitions of concepts | 34 |
| 5.4.2 | Dealing with nested composites | 34 |
| 5.4.3 | Dealing with hardware | 35 |
| 5.4.4 | Pricing of composites | 35 |
| 5.4.5 | Ordering parts | 35 |
| 5.4.6 | Base product and parts indicators | 36 |
| 5.4.7 | LIN-group structure alignment | 36 |
| 6. | IMPLEMENTATION SUGGESTIONS AND RECOMMENDATIONS | 37 |
| 6.1 | STORING AND DISPLAYING THE 7 DIGIT PRODUCT CODE | 37 |
| 6.2 | CHECKING LOT SPECIFICATION RULES AND CONSTRAINTS | 37 |
| 6.3 | STORING AND DISPLAYING MULTIPLE LOT FEATURES | 38 |
| 6.4 | STORING AND DISPLAYING COMPOSITE LOTS | 38 |
| | APPENDIX | 39 |
| I | INVOLVED ORGANISATIONS AND PERSONS | 39 |

1. Introduction

1.1 Background

The Dutch and international flowers and plants business is continuously reinventing itself. Innovation of products, services, business processes and models, as well as technology, permanently put new demands on the volume and depth of data exchange among business partners in the supply chain.

VBN codes make up the core of all data exchange in our business. New developments can't escape touching the VBN code system as well. By now, the current coding practice has stretched the system to its limits. That's why the Linnaeus project was started. Its aim is to suit the system to the demands of the future, such that it meets the requirements of:

- flexibility: it should be easy to add new building blocks
- adaptability: changes should be effected easily and fast
- migratability: the system must be implemented in a stepwise fashion
- cost effectiveness: all this against acceptable costs and effort.

In 2004, a 'Conceptual Blueprint' of the modernized VBN code system was presented. The new approach provided the required functionality:

- detailed specification of products
- table driven control of lot specification rules and constraints
- improved specification of lots
- specification of composite lot components
- logical grouping of features types
- product code list clean-up
- translation of code lists in multiple languages.

Currently, the sector is preparing to implement 'Linnaeus'. To that end, the conceptual blueprint had to be converted into a 'technical blueprint'.

1.2 Purpose

Purpose of this technical blueprint is to provide a complete, precise and unambiguous technical specification of the Linnaeus code system, in particular of the way in which the new system will be reflected in the data exchange among parties, e.g. in code lists and EDI messages.

The contents and structure of code lists and EDI messages described in this document must enable implementation managers, system architects, designers and programmers, to adapt the systems of the involved parties to correctly produce, receive and process the exchanged 'Linnaeus data'.

To software developers, this technical specification will provide information about the impact of the Linnaeus coding system on their business applications. The introduction of Linnaeus will take place in several steps. The exact contents and timing of those steps is described in a separate 'Migration plan'.

The blueprint does not impose imperative guidelines or directives with regards to the realisation within business applications of the involved supply chain partners, but is restricted to giving general suggestions and recommendations.

1.3 Principles

A conceptual model may be technically implemented in several ways. In translating the conceptual blueprint to this technical one, the following principles were followed:

- maximum correspondence: the conceptual model will be mapped to the technical model as faithfully as possible
- minimum redundancy: data will be captured in one single data structure as much as possible, reducing duplication to the minimum
- optimal efficiency: the data structures are designed such as to allow for the most efficient (fast) processing
- minimum impact: the format and contents of existing data structures will be retained as much as possible.

1.4 Contents

In addition to this introduction, this documents contains the following sections:

- Chapter 2: presents Linnaeus' technical data model
- Chapter 3: specifies the structure and contents of Linnaeus code lists
- Chapter 4: deals with the way the code lists are distributed to supply chain partners
- Chapter 5: describes the impact of the Linnaeus code system on the contents and structure of the EDI messages employed by the floricultural industry
- Chapter 6: makes suggestions and recommendations as to the implementation of the Linnaeus system within the business applications of supply chain partners.

An appendix lists the names of persons involved in making this document.

1.5 Status

The present document is the second version of the technical blueprint and the first one in English. Compared to the previous (Dutch) version 2.1, some minor text adjustments and corrections have been made.

Ownership of, and further responsibility for this document rests with VBN. Suggestions for corrections, additions or elucidation of its contents may be submitted to VBN. These suggestions will be evaluated, possibly yielding a new release of the technical blueprint.

1.6 References

For further information about the background of and motives behind the Linnaeus project and its conceptual setup, please refer to the following (Dutch language) documents:

- 'Visieontwikkeling Coderingen in de Sierteeltsector' versie 1.0, July, 1999
- 'Linnaeus Conceptuele Blauwdruk' versie 4.1, April 15, 2005

This technical blueprint is partly based on the following documents:

- 'Linnaeus Ontwerp en Implementatierichtlijnen' versie 3.0, March 4, 2005
- 'Beschrijving Linnaeus/FlorEcom Code listen' versie 0.4, February 24, 2005
- 'Proof of Concept' part 1, February 11, 2005 and 2 of April 8, 2005.

The abovementioned documents may be obtained through VBN or VGB/HBAG.

2. Linnaeus data model

The structure of the Linnaeus coding system is reflected in a 'technical' data model. That data model differentiates between fixed and variable data about floricultural products.

2.1 Fixed data

Fixed data are reference data that will, in principle, be captured in the supply chain partners' business applications only once; such reference data need not be repeatedly included in the information exchange about floricultural business transactions. In the technical data model, these fixed data are represented as blue rectangles.

The administration of fixed reference data is taken care of by VBN, who will secure the correctness, completeness and mutual consistency of the data. The distribution of these fixed data is centrally tended to as well. Hereby, Linnaeus provides floricultural supply chain partners with the opportunity to keep their internal reference data continuously updated and mutually synchronised.

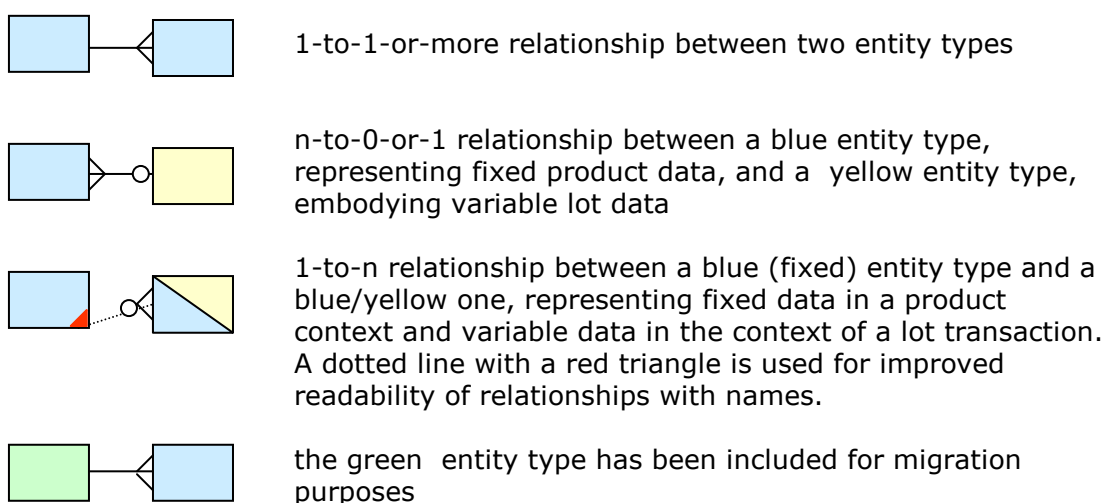
2.2 Variable data

Variable data are attributes of floricultural products which may vary with the lot to which they belong. This data will need to be captured in the database each time a product is featured in a lot and has to be explicitly included in the information exchange about lots at every turn.

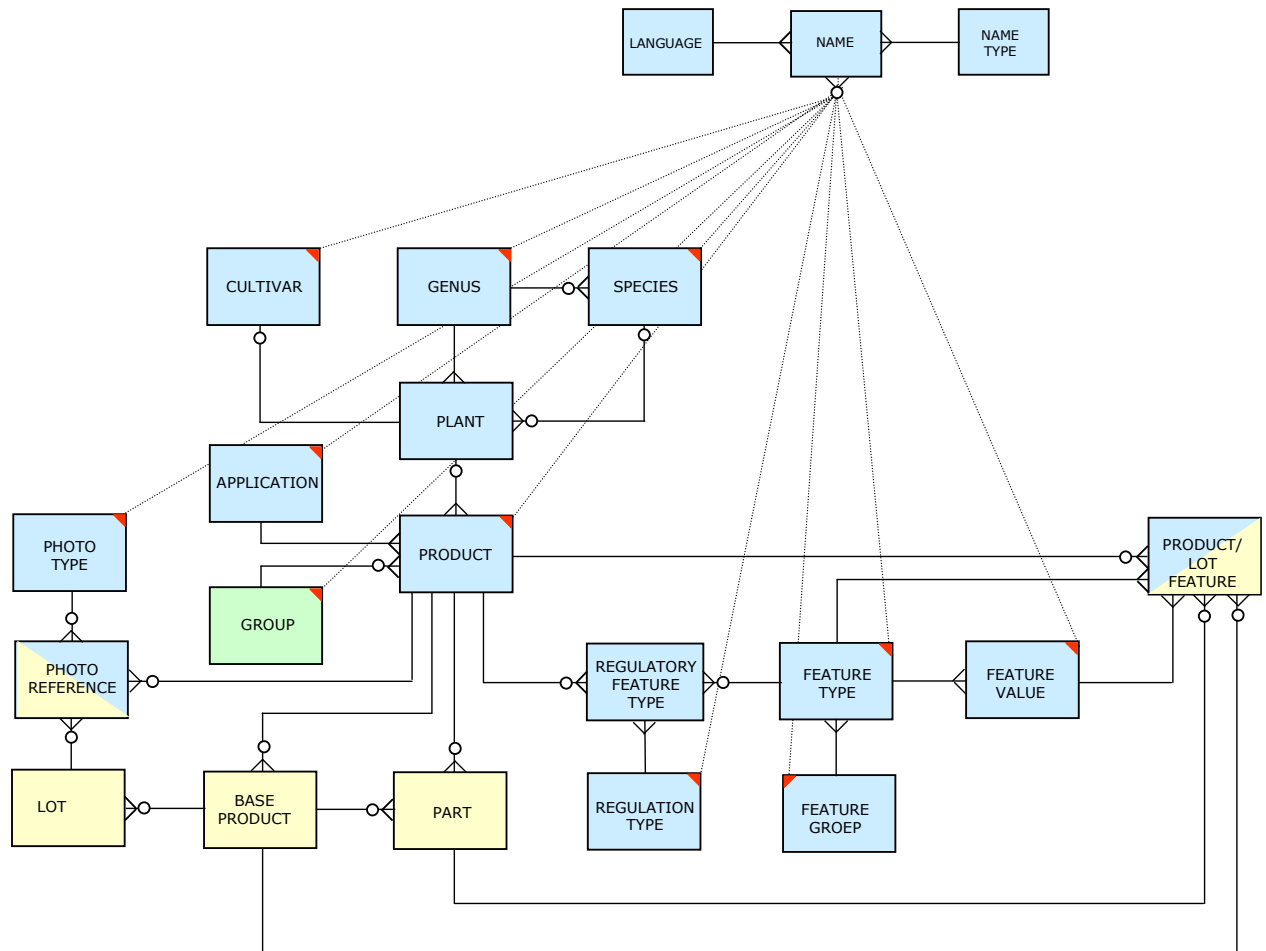
In the technical data model, these variable data are represented as yellow rectangles.

2.3 Data model notation

The model presented below will largely follow the classical entity-relationship notation, presenting entity types as rectangles and relationships as possibly forked line sections:



2.4 Technical data model



The entity types and relationships, belonging to the fixed reference data will be defined and explained in the next section.

Chapter 5 will clarify how the variable transaction data about floricultural lots will be mapped to the FlorEcom EDI messages.

3. Linnaeus code lists

This chapter will describe the code lists used to electronically distribute the fixed Linnaeus reference data.

3.1 Common code list aspects

All code lists exhibit the following generic characteristics.

3.1.1 Field names

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

3.1.2 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 § 4.1 of this document.

3.1.3 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 (except those with *NAME*)
- example of a filled-out code list item (which may differ from reality since VBN may still need to define some codes)
- explanation or remark (if any).

For each code list field the following will be specified:

- field number: field sequence number
- 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, AN = alpha/numeric, number of digits fixed: n, or variable ..n
- key fields: P#: primary key, F#: foreign key, PF#: both primary and foreign key.

3.2 PRODUCT

3.2.1 Definition

A product is a type of good traded within the floricultural industry and identified with a VBN product code. A product may be of a botanical nature (a living or cut plant) or hardware.

3.2.2 Relationships

Product has the following relationships:

- a product always has 1 and only 1 specific application
- a product belongs to 1 and only 1 group
- a product may involve 1, and then only one, type of plant
- a product may have 0, 1 or more product features
- to 1 product 0, 1 or more regulatory feature types may apply
- of 1 product 0, 1 or more photo references may exist.

3.2.3 Contents

The code list *PRODUCT* has the following contents.

| field | field_name | M/C | format | key | contents |
|-------|---------------------------|-----|---------|-----|-------------------------------|
| 1 | code_list_id | M | N..3 | | '1' |
| 2 | product_id | M | N..7 | P# | <i>PRODUCT</i> identifier |
| 3 | application_id | M | N..2 | F# | <i>APPLICATION</i> identifier |
| 4 | VBN_product_name | M | AN..105 | | VBN product name |
| 5 | short_product_name | M | AN..20 | | short product name |
| 6 | registrator_id | C | N..2 | F# | <i>PLANT</i> registrator id |
| 7 | plant_registration_number | C | N..7 | F# | <i>PLANT</i> registration nr |
| 8 | composite_indicator | M | N1 | | '1' or '2' |
| 9 | group_code | M | N8 | F# | VBN group identifier |
| 10 | entry_date | M | N8 | | ccyymmdd |
| 11 | expiry_date | C | N8 | | ccyymmdd |
| 12 | change_date_time | M | N12 | | ccyymmddhhmm |

3.2.4 Example

```
1;10050;2;Calathea roseapicta 'Angela';CALAT ANGELA;1;102421;1;20600301;
20050101;;200412311510
```

3.2.5 Explanation

- 1: code list '1' for *PRODUCT*. This ID allows for the possibility to make an explicit reference to this list from code list *NAME*
- 3: application code, see § **Fout! Verwijzingsbron niet gevonden.**; reference to code list *APPLICATION*
- 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 (clock) display purposes
- 6: ID of the organisation that has registered the plant, for example:
 - 1: VKC
 - 2: VAR-B

Together with field 7, this field constitutes a reference to code list *PLANT*

- 7: 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
- 9: existing VBN group code (included for migration purposes).

3.3 APPLICATION

3.3.1 Definition

Application is the way a product is applied

3.3.2 Relationships

One application will apply to several products.

3.3.3 Contents

The code list *APPLICATION* has the following contents.

| field | field_name | M/C | format | key | contents |
|-------|-------------------------------|-----|--------|-----|-------------------------------|
| 1 | code_list_id | M | N..3 | | '2' |
| 2 | application_id | M | N..2 | P# | <i>APPLICATION</i> identifier |
| 3 | dutch_application_description | M | AN..35 | | description in Dutch |
| 4 | entry_date | M | N8 | | ccyymmdd |
| 5 | expiry_date | C | N8 | | ccyymmdd |
| 6 | change_date_time | M | N12 | | ccyymmddhhmm |

3.3.4 Example

```
2;1;snijproduct;20050101;;200412311510
```

3.3.5 Explanation

- 1: code list number '2' for *APPLICATION*
- 2: application code:
 - 1: snijproduct (cut product)
 - 2: kamerplant (house plant)
 - 3: tuinplant (garden plant)

Possible additional application codes are still to be defined.

- 3: Dutch description of the application.

3.4 PLANT

3.4.1 Definition

A plant is a botanical product of the floricultural industry, taxonomically unambiguously identified with the ID of a VBN acknowledged supplier of plant data. Products with 'other' or 'mixed' in their name are not taxonomically unambiguous so, by definition, these do not count as plants.

3.4.2 Relationships

Plant has the following relationships:

- a plant belongs to 1 and only 1 genus
- a plant may belong to 1, and then only to 1 species
- a plant may be cultivar, and then only 1.

Examples of a plant in relation to genus, species and cultivar are:

- Hedera helix 'Adam': genus with species and cultivar
- Grivillea 'Spiderman': genus with cultivar without species
- Grivillea asplenifolia: genus with species without cultivar.

3.4.3 Contents

The code list *PLANT* has the following contents.

| field_ | field_name | M/C | format | key | contents |
|--------|---------------------------|-----|--------|-----|------------------------------|
| 1 | code_list_id | M | N..3 | | '3' |
| 2 | registrator_id | M | N..2 | P# | <i>PLANT</i> registrator id |
| 3 | plant_registration_number | M | N..7 | P# | <i>PLANT</i> registration nr |
| 4 | genus_id | M | N..5 | F# | <i>GENUS</i> identifier |
| 5 | species_id | C | N..5 | F# | <i>SPECIES</i> identifier |
| 6 | cultivar_id | C | N..7 | F# | <i>CULTIVAR</i> identifier |
| 7 | entry_date | M | N8 | | ccyymmdd |
| 8 | expiry_date | C | N8 | | ccyymmdd |
| 9 | change_date_time | M | N12 | | ccyymmddhhmm |

3.4.4 Example

```
3;1;102421;6003;546;10050;20050101;;200412311510
```

3.4.5 Explanation

- 1: code list number '3' for *PLANT*
- 4: VBN genus code constitutes a reference to code list *GENUS*
- 5: VBN species code (if the plant belongs to a species): reference to code list *SPECIES*
- 6: VBN cultivar code (if the plant is a cultivar): reference to code list *CULTIVAR*.

3.5 GENUS

3.5.1 Definition

A genus is a subdivision of a taxonomical family.

3.5.2 Relationships

Genus has the following relationships:

- a genus may comprise no, one or more species
- a genus may comprise one or more plants.

3.5.3 Contents

The code list *GENUS* has the following contents.

| field_ | field_name | M/C | format | key | contents |
|--------|------------------|-----|--------|-----|------------------|
| 1 | code_list_id | M | N..3 | | '4' |
| 2 | genus_id | M | N..5 | P# | GENUS identifier |
| 3 | latin_genus_name | M | AN..35 | | latin genus name |
| 4 | entry_date | M | N8 | | ccyymmdd |
| 5 | expiry_date | C | N8 | | ccyymmdd |
| 6 | change_date_time | M | N12 | | ccyymmddhhmm |

3.5.4 Example:

4;6003;Calathea;20050101;;200412311510

3.6 SPECIES

3.6.1 Definition

A species is a subdivision of a taxonomical genus.

3.6.2 Relationships

Species has the following relationships:

- a species may comprise no, 1 or more plants
- a species belongs to 1 and only 1 genus.

3.6.3 Contents

The code list *SPECIES* has the following contents.

| field_ | field_name | M/C | format | key | contents |
|--------|--------------------|-----|--------|-----|--------------------|
| 1 | code_list_id | M | N..3 | | '5' |
| 2 | species_id | M | N..5 | P# | SPECIES identifier |
| 3 | genus_id | M | N..5 | F# | GENUS identifier |
| 4 | latin_species_name | M | AN..35 | | latin species name |
| 5 | entry_date | M | N8 | | ccyymmdd |
| 6 | expiry_date | C | N8 | | ccyymmdd |
| 7 | change_date_time | M | N12 | | ccyymmddhhmm |

3.6.4 Example

5;546;6003;roseapicta;20050101;;200412311510

3.7 CULTIVAR

3.7.1 Definition

Cultivar is a specialisation of a botanical species or genus at the lowest possible taxonomical level.

3.7.2 Relationship

A cultivar may comprise one or more plants.

3.7.3 Contents

The code list *CULTIVAR* has the following contents.

| field_ | field_name | M/C | format | key | contents |
|--------|------------------|-----|--------|-----|---------------------|
| 1 | code_list_id | M | N..3 | | '6' |
| 2 | cultivar_id | M | N..7 | P# | CULTIVAR identifier |
| 3 | cultivar_name | M | AN..35 | | cultivar name |
| 4 | entry_date | M | N8 | | ccyymmdd |
| 5 | expiry_date | C | N8 | | ccyymmdd |
| 6 | change_date_time | M | N12 | | ccyymmddhhmm |

3.7.4 Example

6;10050;'Angela';20050101;;200412311510

3.7.5 Explanation

- 3: the name under which a cultivar is traded. If the trade name differs from the official botanical name, the latter name will be included in the code list *NAME* with name type '3' see § 3.14.5.

3.8 PRODUCT FEATURE

3.8.1 Definition

A product feature is a fixed property of a product, expressed as the value of a feature type. Examples of product features are:

- bloemkleur: rood (flower colour: red)
- steellengte: 20 cm (stem length: 20 cm)
- gewasgroep: Heesters (plant group: Shrubs)
- CBS groep: 06031010 (CBS group: 06031010).

3.8.2 Relationships

Product feature has the following relationships:

- a product feature involves 1 and only 1 feature type
- a product feature has 1 and only 1 feature value
- a product feature pertains to 1 and only 1 product.

3.8.3 Contents

The code list *PRODUCT_FEATURE* has the following contents.

| field_ | field_name | M/C | format | key | contents |
|--------|------------------|-----|--------|-----|--------------------|
| 1 | code_list_id | M | N..3 | | '7' |
| 2 | product_id | M | N..7 | PF# | PRODUCT identifier |
| 3 | feature_type_id | M | AN3 | PF# | FEATURE_TYPE id |
| 4 | feature_value | M | AN..3 | PF# | FEATURE_VALUE id |
| 5 | entry_date | M | N8 | | ccyymmdd |
| 6 | expiry_date | C | N8 | | ccyymmdd |
| 7 | change_date_time | M | N12 | | ccyymmddhhmm |

3.8.4 Example

7;9156;S50;010;20050101;;200501021510

3.8.5 Explanation

- 3: the feature type id filed will always contain 3 alpha numerical digits
- 4: feature values fields will contain at most 3 alpha numerical digits.

3.9 FEATURE TYPE

3.9.1 Definition

A feature type defines the type of property to which a product features pertains. Examples of feature types are:

- bloemkleur (flower colour)
- steellengte (stem length)
- gewasgroep (plant group)
- CBS group (CBS group).

3.9.2 Relationships

Feature type has the following relationships:

- a feature type may have 1 or more feature values
- a feature type will belong to 1 and only 1 feature group.

3.9.3 Contents

The code list *FEATURE_TYPE* has the following contents.

| field | field_name | M/C | format | key | contents |
|-------|--------------------------------|-----|--------|-----|-------------------------|
| 1 | code_list_id | M | N..3 | | '8' |
| 2 | feature_type_id | M | AN3 | P# | <i>FEATURE_TYPE</i> id |
| 3 | feature_group_id | M | N..3 | F# | <i>FEATURE_GROUP</i> id |
| 4 | dutch_feature_type_description | M | AN..35 | | dutch description |
| 5 | entry_date | M | N8 | | ccyymmdd |
| 6 | expiry_date | C | N8 | | ccyymmdd |
| 7 | change_date_time | M | N12 | | ccyymmddhhmm |

3.9.4 Example

8;S01;2;Potmaat;20050101;;200501021510

3.10 FEATURE VALUE

3.10.1 Definition

A feature value encodes a specific product property of a certain type. Examples of feature values are:

- Rood (red)
- 20 cm (20 cm)
- Heesters (Shrubs)
- 06031010 (06031010).

3.10.2 Relationships

The relationships of feature value are the inverse of the relationships already defined for feature and feature type with this entity type.

3.10.3 Contents

The code list *FEATURE_VALUE* has the following contents.

| field | field_name | M/C | format | key | contents |
|-------|---------------------------------|-----|--------|-----|-------------------------|
| 1 | code_list_id | M | N..3 | | '9' |
| 2 | feature_type_id | M | AN3 | PF# | <i>FEATURE_TYPE</i> id |
| 3 | feature_value | M | AN..3 | P# | <i>FEATURE_VALUE</i> id |
| 4 | dutch_feature_value_description | M | AN..35 | | dutch description |
| 5 | entry_date | M | N8 | | ccyymmdd |
| 6 | expiry_date | C | N8 | | ccyymmdd |
| 7 | change_date_time | M | N12 | | ccyymmddhhmm |

3.10.4 Example

9;S01;010;10 CM POT;20050101;;200501021510

3.10.5 Remark

All feature values, exclusively consisting of numeric characters, are filled out to 3 positions using preceding zeroes. Some feature values are auction specific; if so, these will not be included in and distributed through this code list.

3.11 FEATURE GROUP

3.11.1 Definition

A feature group is a set of feature types concerning a common aspect. To date, the following feature groups are proposed:

- botanical features
- sorting features
- order features
- transport features
- auction features
- article features
- quality features
- transaction features
- export features
- other features.

3.11.2 Relationships

A feature group will comprise 1 or more feature types.

3.11.3 Contents

The code list *FEATURE_GROUP* has the following contents.

| field | field_name | M/C | format | key | contents |
|-------|---------------------------------|-----|--------|-----|-------------------------|
| 1 | code_list_id | M | N..3 | | '10' |
| 2 | feature_group_id | M | N..3 | P# | <i>FEATURE_GROUP</i> id |
| 3 | dutch_feature_group_description | M | AN..35 | | dutch description |
| 4 | entry_date | M | N8 | | ccyymmdd |
| 5 | expiry_date | C | N8 | | ccyymmdd |
| 6 | change_date_time | M | N12 | | ccyymmddhhmm |

3.11.4 Example:

10;4;Quality features;20050101;;200501021510

3.12 REGULATORY FEATURE TYPE

3.12.1 Definition

Regulatory feature types are types of feature belonging to a product which should of may be specified on certain prescribed presentation positions. Therefore the regulation of feature types consists of 3 elements: feature type, regulation type and presentation order.

3.12.2 Relationships

Regulatory feature type has the following relationships:

- a regulatory feature type pertains to 1 and only 1 product
- a regulatory feature type involves 1 and only 1 feature type.
- a regulatory feature type involves 1 and only 1 regulation type.

3.12.3 Contents

The code list *REGULATORY_FEATURE_TYPE* has the following contents.

| field | field_name | M/C | format | key | contents |
|-------|--------------------|-----|--------|-----|----------|
| 1 | code_list_id | M | N..3 | | |
| 2 | product_id | M | N..7 | PF# | |
| 3 | feature_type_id | M | AN3 | PF# | |
| 4 | regulation_type_id | M | N..2 | PF# | |
| 5 | presentation_order | M | N..2 | P# | |
| 6 | entry_date | M | N8 | P# | |
| 7 | expiry_date | C | N8 | | |
| 8 | change_date_time | M | N12 | | |

3.12.4 Example

```
11;9152;S01;1;1;20050101;;200501021510
```

3.12.5 Explanation

2,3,4,5: as soon as in a combination of product_id and feature_type_id one or more fields (regulation_type_id, presentation_order) change, a new line with a new entry_date comes in to being. In the "old line" the expiry date will be indicated.

4: the regulation type code indicates whether the involved feature type is mandatory, conditional, advised or permitted. The adoption of these various regulation types still is subject to discussion. Possible implementation:

- 1: verplicht (mandatory)
- 2: geadviseerd (advised)
- 3: toegestaan (permitted)
- 4: conditioneel (conditional) see § 6.2

5: the position in the order in which a feature value is presented on those media that, due to lack of space, can't display the feature type.

3.12.6 Business rules

Both of the following business rules apply:

- on one moment a feature type for one product can not be used more than once.
- on one moment a presentation position for one product can not be used more than once.

3.13 REGULATION TYPE

3.13.1 Definition

A regulation type defines the scope of a regulation.

3.13.2 Relationships

A regulation type may pertain to various regulatory feature types.

3.13.3 Contents

The code list *REGULATION_TYPE* has the following contents.

| field_ | field_name | M/C | format | key | contents |
|--------|-----------------------------------|-----|--------|-----|---------------------------|
| 1 | code_list_id | M | N..3 | | '12' |
| 2 | regulation_type_id | M | N..2 | P# | <i>REGULATION_TYPE</i> id |
| 3 | dutch_regulation_type_description | M | AN..35 | | dutch description |
| 4 | entry_date | M | N8 | | ccyymmdd |
| 5 | expiry_date | C | N8 | | ccyymmdd |
| 6 | change_date_time | M | N12 | | ccyymmddhhmm |

3.13.4 Example

```
12;1;verplicht;20050101;;200501021510
```

3.14 NAME

3.14.1 Definition

A name is

- a description of an item in a language other than the standard language
- an alternative name, or
- an abbreviation of an item's standard or translated description.

For botanical names, Latin is the standard language; for all other descriptions, the standard language is Dutch.

3.14.2 Relationships

Name has the following relationships:

- a name applies to 1 and only 1 occurrence of a certain entity type (product, cultivar, genus, species, application, feature type, feature value, feature group, photo type or regulation type)

- an occurrence of a certain entity type may have 0, 1 or more different names
- a name is put in 1 and only 1 language
- a name involves 1 and only 1 name type.

3.14.3 Contents

The code list *NAME* has the following contents.

| field | field_name | M/C | format | key | contents |
|-------|--------------------------|-----|---------|-----|-----------------------------|
| 1 | code_list_id | M | N..3 | | '13' |
| 2 | involved_code_list_id | M | N..3 | P# | code list identifier |
| 3 | code_list_item_id | M | AN..14 | PF# | code list key field |
| 4 | second_code_list_item_id | C | AN..7 | PF# | 2nd code list key field |
| 5 | name_type_id | M | N..2 | PF# | <i>NAME_TYPE</i> identifier |
| 6 | language_id | M | AN2 | PF# | <i>LANGUAGE</i> identifier |
| 7 | name_or_translation | M | AN..255 | | name or translation |
| 8 | entry_date | M | N8 | | ccyymmdd |
| 9 | expiry_date | C | N8 | | ccyymmdd |
| 10 | change_date_time | M | N12 | | ccyymmddhhmm |

3.14.4 Example

```
13;2;2;;1;en;house plant;20050101;;200501021510
```

3.14.5 Explanation

- 2: number of the code list containing the named item
- 3: (first) key value (ID) of the named item in the involved code list
- 4: if the item involves a feature value, then this field will specify the second key value (feature_value) making up the item's primary key
- 5: name type: suggested values (still to be decided):
- 1: translation
 - 2: alternative name
 - 3: official name (formal taxonomical plant name)
 - 4: common name (popular name, when supported)
 - 5: abbreviation
- 6: ISO 639 code of the language in which the name is written; in case of formal botanical names, the language code 'la' for Latin is used.
- 7: name, description or abbreviation of the involved item in the indicated language.

All items in code lists accompanied by reference data (when this is deemed relevant) will be provided with abbreviations, possibly in more than one language. The items included in some code lists will be provided with an alternative name. These alternative names, accompanied whenever necessary by their names in languages other than Dutch (referred to here as "support languages"), will also be managed and distributed from a central location. The introduction of multilingual Linnaeus code lists will be done in two steps. During step 1, beginning on 1 January 2007, English, German and French will be added to the Dutch. The relevant abbreviations in the support languages will also become available during this phase. Once the code lists have been evaluated, they can start being translated into other languages and managed and distributed from a central location. The translation will start on 1 January 2009 or earlier if possible.

3.15 NAME TYPE

3.15.1 Definition

A name type defines what a name denotes and what's its intent.

3.15.2 Relationships

Inverse of the relationships between name and name type described in the previous section.

3.15.3 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 | | '14' |
| 2 | name_type_id | M | N..2 | P# | <i>NAME_TYPE</i> identifier |
| 3 | english_name_type_description | M | AN..35 | | name type in English |
| 4 | entry_date | M | N8 | | ccyymmdd |
| 5 | expiry_date | C | N8 | | ccyymmdd |
| 6 | change_date_time | M | N12 | | ccyymmddhhmm |

3.15.4 Example

14;3;Official name;20050101;;200501021510

3.15.5 Explanation

- 2: code of the name type
- 3: English description of the name type.

3.16 LANGUAGE

3.16.1 Definition

A language is the way in which members of a language community express thoughts and notions.

3.16.2 Relationships

One particular language may serve to express several names.

3.16.3 Contents

The code list *LANGUAGE* has the following contents.

| field_ | field_name | M/C | format | key | contents |
|--------|-----------------------|-----|--------|-----|----------------------------|
| 1 | code_list_id | M | N..3 | | '15' |
| 2 | language_id | M | AN2 | P# | <i>LANGUAGE</i> identifier |
| 3 | english_language_name | M | AN..35 | | language in English |
| 4 | entry_date | M | N8 | | ccyymmdd |
| 5 | expiry_date | C | N8 | | ccyymmdd |
| 6 | change_date_time | M | N12 | | ccyymmddhhmm |

3.16.4 Example

15;en;English;20050101;;200501021510

3.16.5 Explanation

- 2: ISO 639 2-digit language code
3: English name of the language.

3.17 GROUP

3.17.1 Definition

A group is how VBN products were classified before the introduction of Linnaeus.

3.17.2 Relationship

A group consists of 0, 1 or more products. The GROUP code list contains groups on level 1, 2, 3 or 4. Only groups on level 4 relate with PRODUCT.

3.17.3 Contents

The *GROUP* code list has the following contents.

| field | field_name | M/C | format | key | contents |
|-------|-------------------------|-----|--------|-----|-------------------------|
| 1 | code_list_id | M | N..3 | | '16' |
| 2 | group_code | M | N8 | P# | VBN group identifier |
| 3 | dutch_group_description | M | AN..35 | | Dutch group description |
| 4 | entry_date | M | N8 | | ccyymmdd |
| 5 | expiry_date | C | N8 | | ccyymmdd |
| 6 | change_date_time | M | N12 | | ccyymmddhhmm |

3.17.4 Example

16;10000000;Snijsbloemen;19800101;;200410191514
16;10300000;Knol- en bolbloemen;19930809;;199512281206
16;10300500;Freesia;19951220;;199520121426
16;10300502;Freesia dubbel;19951220;;200203061046

3.17.5 Explanation

In time, this code list will be removed from the distribution

3.18 PHOTO REFERENCE

3.18.1 Definition

A photo reference is an identification and specification of a digital image belonging to a product or a lot.

3.18.2 Relationships

Photo reference has the following relationships:

- if a photo reference involves a product, it will concern 1 and only 1 product
- if a photo reference involves a lot, it will concern 1 and only 1 lot
- a photo reference has 1 and only 1 photo type.

3.18.3 Contents

The precise contents of the *PHOTO_REFERENCE* code list is still to be decided and its specification will be included in a coming release of this document.

3.19 PHOTO TYPE

3.19.1 Definition

Photo type specifies the type of a photo.

3.19.2 Relationships

One photo type may apply to several photo references.

3.19.3 Contents

The precise contents of the *PHOTO_TYPE* code list is still to be decided and its specification will be included in a coming release of this document..

4. Code list distribution

This section will discuss policy matters and IT aspects of the Linnaeus code list distribution.

4.1 Distribution policy

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

4.1.1 Responsibilities

The contents of Linnaeus code lists is maintained by the involved code list agencies. These organisations are responsible for the correctness, completeness and the mutual consistency of the data.

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

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

4.1.2 Complete set

A basic tenet of the Linnaeus code lists distribution is that any business partner, at any time, must be able to obtain the complete set of Linnaeus code lists from the distribution agent, including possibly expired codes which make up a reference to historical data.

The starting point for the distribution of code lists is that it should be possible at any time for every partner in the chain to obtain the complete set of Linnaeus code lists, along with any future codes and any blocked codes that refer to historic data, as a cohesive unit from the distribution system. This set of data, to which the technical blueprint applies, concerns the set starting with the letter F, i.e., the F files.

In addition to the F files, C files will be delivered. The C files will contain a selection taken from the F files ("C" standing for "current" and "F" standing for "full"). The C files will contain the product codes, with related items, for which the deadline has not yet been reached and for which the effective date is previous to or the same as the day after tomorrow. Items that become effective in the future and are included in the F files will thus not appear in the C file until the time when the effective date is the same as the day after tomorrow. Items having a deadline and that are included in the F files will thus be removed from the C file as soon as their deadline has passed.

For example:

| F-file | C-file |
|---------------------------------|---------------------------------|
| Effective > 2 days | |
| Currently effective + 2 days | Currently effective + 2 days |
| Blocked | |

4.1.3 Update detection

Another principle is that new, expired and changed items are to be detected easily within the complete data set, 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 § 3.1.2).

To establish which codes 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
- expired: change_date_time is later than the last processing date/time and expiry_date is later than or equal to the change_date
- changed: change_date_time is later than the last processing date/time and entry_date is before the change_date.

To warrant a proper functioning of this procedure, VBN and FlorEcom will ensure that:

- expired items will be permanently included in the code lists, with the exception of the items mentioned after the first bullet of § 4.1.4
- 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. It is possible, however, to reactivate items such as a product code. In such a case, a previously distributed expiration date would be removed.

4.1.4 Special cases in the detection of updates

In the process of detecting updates, the following special cases may occur:

- expired items, belonging to product related code lists ('PRODUCT_FEATURE', 'REGULATORY_FEATURE_TYPE', 'NAME' and 'PHOTO_REFERENCE'), or other items that have expired to correct

erroneous entry, might not necessary remain included in the code lists permanently, but may be left out after some time

- users with a low update frequency may fail to detect new items as being 'new' if those new items are changed shortly afterwards, but they will be detect as being 'changed'
- for technical reasons, it's theoretically possible that items which have been notified as being 'expired' may yet undergo changes, and will then be detected as being 'changed'.

Data should be updated when approving and implementing the procedure:

- data that has been entered incorrectly or twice can occasionally be removed from the code lists.
- although the data associated with an expired product will remain permanently included in the full list, products that expired more than 7 years ago and the data related to these products will be removed annually.

4.1.5 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 <i>in the database of the code list responsible agency</i> ; 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 (e.g.: a change in the regulation type of a regulatory feature type) 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.

4.2 Distribution technique

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

4.2.1 File format

The code list files will be available as CSV files, compressed in .zip format, using:

- semicolon as field separator
- CR-LF as record separator.

The files will be sorted on primary key. If so desired by the users, the distribution agent may offer the files in other formats, such as HTML, EDI, XML, XLS or any other. It's also conceivable that FlorEcom will provide a (web)service, providing update detection on behalf of the users, so as to minimize data traffic over the network. In the event, this service will be elaborated by FlorEcom later.

4.2.2 Character set

Since the *NAME* 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 Ediflower/FlorEcom EDI messages, 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.

Special characters in product names

Some applications in the supply chain, like most auction clocks, are unable to display diacritic characters. This implies that product names with special characters may not be correctly displayable on a clock front. VKC (a plant registration authority) has rules allowing a limited set of diacritic characters, meaning that plant names with special characters may occur.

4.2.3 Distribution channel

The code list responsible agents will put their data in the appropriate directories on the FlorEcom server using the FTP protocol. Thence, the lists will be made available to the FlorEcom business partners in two ways:

- FlorEcom website: a web page allowing the user to manually download the code lists
- FlorEcom FTP site: an FTP site, accessible by the user or an application using an FTP client or a Web browser, providing access to the Linnaeus code list directories on the basis of a login-ID, user name and password.

The following applies with regards to:

Security

The Linnaeus code lists are of a (semi) public nature. To keep some control over the distribution platform's payload and performance, the code list will be made available to users with a login-ID and password only. ID and password will not be exchanged in an encrypted fashion.

Further details about the distribution technique will be conveyed by the distribution agent in a later stage.

5. Linnaeus in EDI messages

This chapter will deal with the impact of introducing the Linnaeus code system on the contents and structure of the EDI messages applied in the floricultural industry.

5.1 Mapping the Linnaeus technical data model on EDI messages

The technical data model of the Linnaeus code system defines the following entity types related to the data exchange about floricultural lots.

5.1.1 Lot

Definition: A lot is a number of products having exactly the same features on the level of the sales unit, which are being marketed as a whole, on one specific place and one specific point in time, the legal ownership of which rests with one and only one supply chain partner.

Relationships: Lot has the following relationships:

- a lot has 1 and only 1 base product
- a lot may be pictured on one or more photos or none (have 0, 1 or more photo references).

Mapping: In an EDI message, a lot is mapped on the LIN group.

5.1.2 Base product

Definition: A base product is a lot's primary product.

Relationships: Base product has the following relationships:

- a base product refers to 1 and only 1 product
- a base product may have 0, 1 or more parts
- a base product may have 0, 1 or more lot features.

Mapping: In an EDI message, the base product is specified in the LIN main line.

5.1.3 Part

Definition: A part is a product which is a component of, or an accessory to a base product.

Relationships: Part has the following relationships:

- a part belongs to 1 and only 1 product
- a part may have 0, 1 or more lot features.

Mapping: In an EDI message, a part is specified in the LIN sub line. For further explanation of the notions 'base product' and 'part', see § 3.5 and 3.6 van Conceptual Blueprint (in Dutch).

5.1.4 Lot feature

Definition: A lot feature is a feature of a base product or of a part of a lot, expressed in terms of the value of a feature type.

Relationships: Lot feature has the following relationships:

- a lot feature involves 1 and only 1 feature type
- a lot feature involves 1 and only 1 feature value
- a lot feature pertains to 1 or more base products or parts.

Example: Examples of lot features are:

- verkoopenheid: bos (sales unit: bunch)
- potmaat: 14 cm (pot size: 14 cm)
- aantal stekken: 4 (number of cuttings: 4)
- fytoosanitair kenmerk: 100% Japanse roest vrij (phytosanitary feature: 100% free of Japanese rust)

Mapping: In an EDI message, lot features will be specified in the LIN.CCI-CAV group; feature type in LIN.CCI and feature value in LIN.CAV.

5.1.5 Photo reference

Definition: A photo reference is the identification of a digital image belonging to a certain lot.

Relationships: Photo reference has the following relationships:

- a lot may have 0, 1 or more lot photo references
- a lot photo reference may pertain to 1 and only 1 lot

Mapping: In an EDI message, photo references are (possibly repeatedly) specified in a LIN.RFF segment.

5.2 Extension of lot features

One of the most obvious consequences of introducing Linnaeus will be a considerable increase of the number of lot features.

In the FlorEcom messages these features will be specified in the CCI-CAV group. The number of occurrences of that group is currently limited to 20. In some messages, notably Ediflower EAB and EKT, lot features are not specified in CCI-CAV but in the IMD segment, which may occur maximum 30 times.

The 'proof of concept' by the Linnaeus project group has demonstrated that, in actual practice, a number of 20 lot features is quite realistic. In the most recent version D04B of the Edifact standard, the maximum number of CCI-CAV group is 999. The Linnaeus technical implementation task force therefore proposes to raise the number of CCI-CAV group occurrences in the new FlorEcom/Ediflower message releases to 999.

5.3 Harmonisation of the EDI messages

The final elaboration of below recommendations with regards to the harmonisation of floricultural EDI messages will be FlorEcom's responsibility.

5.3.1 Uniform specification of lot features

In the interest of a uniform handling of lot features by processing applications, it is important that all EDI messages deal with features in a similar fashion. Future releases of the Ediflower/FlorEcom EKT and EAB messages will therefore introduce the CCI-CAV group, replacing IMD to specify lot features.

5.3.2 Uniform field formats

The present floricultural EDI messages aren't fully consistent in their definition of identifier field formats. Some identifiers are defined as numeric field while others are alpha numeric. In addition, the use of heading zero's and trailing spaces isn't fully clear.

In view of the key role of identifiers in the processing of EDI messages, precise definitions of identifier field formats are highly important. That's why it is proposed that identifiers within the context of Linnaeus and Ediflower/FlorEcom are uniformly defined as numeric fields of variable length (N..n). Exception to this rule are the following identifiers:

- Feature types: AN3
- Feature values¹: AN..3
- Lot identifiers: AN..15
- Load carrier RFID's: AN..26
- EAN numbers: N13
- ISO language codes: AN2
- ISO country codes: AN2

In numeric fields, heading zero's are redundant; in EDI message exchange such zero's will be eliminated by standard Edifact translators. Likewise, such translators will remove possible trailing spaces in alphanumeric fields.

5.3.3 Extension of the VBN product code

The extension of the VBN product code from 5 to 7 digits calls for a change of the LIN 7140 data-element format from AN..5 to N..7 in *all* existing FlorEcom/Ediflower messages.

¹

The format of most feature values currently is, and will remain, AN3. The precise definition of these values will be decided upon by VBN, taking as the rule that numeric feature values will be filled out to precisely 3 position using preceding zeroes if needed. Exception to this rule are values of feature types S69 and S98 whose format will remain AN2

5.4 Specification of composite lots

The conceptual blueprint of the Linnaeus code system acknowledges the benefits of supporting composite lots. It's description of how composite lots are specified is largely based on the current implementation of composites in the existing FlorEcom messages, only to propose a some minor changes to counter a few sticking points in the present specification.

If, when and to what extend the proposed specification of composite lots will be adopted as a standard within the industry, is still to be decided. If so decided, however, it is evident that this will result in changes of the current FlorEcom convention manual, notably on the following counts:

- definition of the notions of accessory, option and part
- multiply nested composites
- dealing with hardware
- pricing of composites
- ordering of parts
- indicating base product and parts
- harmonisation or the LIN-group structure.

5.4.1 Definitions of concepts

The current FlorEcom convention manual employs the following notions related to the composite lot phenomenon:

- product: the product which is being specified in the message LIN segment, acting as a 'master line'
- accessory: an optional addition to a product, specified in a LIN segment, acting as a 'sub line' to a master line'
- part: a permanent part of a product that is specified, without its own price, in a LIN segment acting as a 'sub line' to its 'master line'.

In future releases of the FlorEcom messages, the above notions will be replaced by:

- base product: the 'main' product of a lot which is specified in an EDI message LIN 'base line'
- optional part: a single (non-composite) product that may freely be ordered as an accessory to a composite base product, specified in the LIN segment acting as 'sub line' to its 'base line'
- fixed part: a single (non-composite) product constituting a permanent, not separately orderable component of a composite base product, specified without a price in the LIN segment acting as 'sub line' to its 'base line'.

5.4.2 Dealing with nested composites

The current FlorEcom messages allow composite products to be part of a composite themselves. Such multiple nesting has proven to complicate displaying composite products in most environments.

In the newly proposed setup, composites may only act as base products, never as a part of another composite. This implies that parts of composite products (like mixed trolleys), may only be specified at the lowest level (of individual plant or possibly hardware). The drawback of this approach, being the inability to establish whether the packages on the trolley are composite items themselves and how they are composed, is resolved by specifying feature values of type S26 (mixed per trolley), S27 (mixed per shelf) and/or S28 (mixed per package).

To indicate how the products are mixed inside a package, it is proposed to introduce a separate feature type 'package sorting' with values such as 'sorted per row', 'diagonally sorted' or 'randomly packed'.

5.4.3 Dealing with hardware

The current FlorEcom convention manual is unclear as to how hardware parts of composite product lots are specified. On the one hand, hardware may be specified using S61 feature type values. On the other, the convention allows for hardware to be indicated in sub lines, in which case specificity is severely limited by the small number of available VBN code for hardware (codes '2001' and '1999' only).

The new approach assumes that VBN codes will be assigned to a limited number of about 40 hardware categories. Hardware parts will henceforth be specified using the appropriate VBN code in the sub line LIN segment rather than through S61 feature type values in a CCI-CAV group².

5.4.4 Pricing of composites

Currently, partly due to the ambiguity in specifying parts mentioned above, the FlorEcom messages fail to be fully clear about the exact price of a composite item. The proposed new convention avoids this ambiguity by applying the following rules:

- the price of a composite item is equal to the price of its base product and the sum of the prices of its optional parts
- the price of an optional part must always be specified, in the LIN.PRI segment of a message; if an optional part is free of charge, its price is specified to be 0
- the price of a fixed part is considered to be included in the price of the composite; fixed parts never have a price.

5.4.5 Ordering parts

At present, FlorEcom allows buyers to specify the quantities and features of composite parts as they please. This has been causing misinterpretation and potential disruptions in automatic transaction handling. The new Linnaeus approach will avoid these problems by applying the following rules:

- in the ordering process of composite lot products, one may indicate the number of optional parts but not their features
- if, when ordering composite lot products, an optional part has not been explicitly included, than that part is considered not to be desired
- when ordering composite lot products, the number of fixed parts and their features are considered given, they may not be changed.

² Some Edifact messages do not yet support CCI-CAV for specifying lot features, in which case IMD is employed

5.4.6 Base product and parts indicators

In the current FlorEcom messages the type of a composite part is indicated by means of a so called sub line indicator ('1' for 'part' and '2' for 'accessory'). An indicator for base products does not currently exist.

The new implementation of composite lot products in the FlorEcom messages will support sub line indicators reflecting the revised definitions as discussed in § 5.4.1:

- 1: fixed part
- 2: optional part.

In order to explicitly (but in fact redundantly) indicate that a LIN-group pertains to a base product or a part, it is proposed to require the LIN data-element 1222: configuration level, to be filled with the following mandatory values:

- 1: if a LIN-group involves a base product
- 2: if a LIN-group involves a part.

5.4.7 LIN-group structure alignment

The present FlorEcom convention manual discerns different types of LIN-group structures: for base products, accessories and parts. These different structures complicate the process of testing and certifying by FlorEcom. Moreover, in practice the structural differences between base products and parts are small.

That's why it is proposed to iron out the differences between base line and sub line structures in the FlorEcom message and make these uniform instead. Practically this will come down to defining some currently mandatory segments within the LIN-group as conditional in future versions.

6. Implementation suggestions and recommendations

This chapter will include suggestions and recommendations with respect to the implementation of the Linnaeus system in business applications of the involved supply chain partners.

6.1 Storing and displaying the 7 digit product code

Product codes form the core of floricultural supply chain partners' business applications. Anywhere these applications store, show or print the currently 5 digit product code, it will have to reckon with the possibility of a 7 digit code. The effects of this will be felt in particular in applications where the number of positions is fixed and/or limited, like for instance on:

- character based computer screens
- special displays (clock fronts hand terminals)
- graphical interfaces that try to avoid scrolling
- print-outs (dispatch letters, distribution notes, labels and stickers, invoices, reports etc.).

One of the aims of a Linnaeus 'proof of concept' was to assess, in broad terms and not necessarily exhaustively, which sub systems of the auction and trade applications were likely to be affected by the extension of the number of product code digits and lot features. For detailed information about this impact, we refer to the involved document (available in Dutch only).

6.2 Checking lot specification rules and constraints

The floricultural sector is highly attached to a strict observance of regulations regarding lot specification. The business partners' applications may play an important role in enforcing these regulations, notably when:

- (new) lots are manually entered
- electronic messages containing lot information are being received.

Based on the contents of the regulatory feature type code list for a particular lot product, the application could:

- see to it that all mandatory feature types are filled out
- enforce that conditional feature types are completed
- indicate which feature types are advised
- check that disallowed feature types are left out³
- verify that only valid feature type values are used.

The regulatory feature types code list may specify that a particular feature type will be conditional, but it won't spell out what these conditions are. These concrete conditions are included in the paper version of the product specifications. It's up to the developers of business applications to build in these conditions into their applications to be called and evaluated in case of conditional feature types.

³

Whether or not the occurrence of disallowed feature types can indeed be checked, will depend on a still to be taken decision about the inclusion of 'allowed' feature types..

6.3 Storing and displaying multiple lot features

Inasmuch business applications still have lot features implemented as fixed attributes, the extension of the potential number of lot features will entail the introduction of a new and distinct 'lot feature' table, reflecting a 1-to-many relationship between lot and features.

Whenever lot data is shown on a computer screen, the obvious solution would seem to display the mandatory lot features in a limited number of fixed fields, while the other features are presented on demand by means of an interactively callable pop-up window.

6.4 Storing and displaying composite lots.

The introduction of composite lots, if and when decided upon, will leave the existing lot data structures within an application basically untouched. The current lot data matches the new base product data; both are captured in a FlorEcom message base line.

Component parts are likely to be stored in a newly introduced 'parts' table, implementing a 1-to-many relationship between lot and parts. The parts data structure resembles the lot table in its being identified by a VBN code and further specified by an arbitrary number of features. Just like lot base product features, the specification of part features is governed by the contents of the regulatory feature type table.

An obvious approach to displaying composites on a computer screen is to fit such lot products with an composite indicator, allowing the user to interactively call a pop-up window showing its component parts.

Appendix

I *Involved organisations and persons*

The persons mentioned below have directly or indirectly contributed to the realisation of this technical blueprint:

- Fred van Andel (VBA)
- Bert van den Bosch (FloraHolland)
- Bas de Bruyn (FloraHolland)
- Ben Ensink (HBAG)
- Gerhard Heemskerk (DD/FlorEcom)
- Arie van Heusden (GreenSystems)
- Jan Holtkamp (Linnaeus projectleider)
- Frank Koldijk (Koldijk Consultants)
- Christo van der Meer (VBA)
- Nico van der Meer (VBA)
- Ron Nieboer (VON)
- Arie Oud (VON)
- Jan Ratterman (VBA)
- Nico Reichenbach (VBN)
- Heino van Rijnberk (VBN)
- Fred Sanders (VBA)
- Irene Sinteur (VBN)
- Wout Verheij (Anteater)
- Henk Zwinkels (FlorEcom).