Sen4CAP - Sentinels for Common Agricultural Policy

Design Justification File

ATBD for the Subsidy Application Layer Preparation

Milestone	Milestone 2
Authors	Philippe Malcorps, Nicolas Bellemans, Sophie Bontemps, Laurentiu Nicola
Distribution	ESA - Benjamin KOETZ

This page is intentionally left blank

Table of contents

List of figures

List of tables

List of acronyms

Acronym	Definition
S1	Sentinel-1
S2	Sentinel-2
PA	Paying Agency
LPIS	Land Parcel Identification System
GSAA	GeoSpatial Aid Application
NLD	The Netherlands
LUT	Look-Up Table

1. Logical model and processor overview

In order to ensure a certain level of consistency between the different Sen4CAP processors (L4A crop type, L4B grassland mowing detection and L4C agricultural practices monitoring), it is needed to standardize the original subsidy application layer received from each Paying Agency (PA). This is the first goal of the subsidy application layer preparation. The second goal is to qualify each parcel with a set of indicators or flags related to their geometry and area, and the third goal is to provide intermediate products based on the subsidy application layer, which are used by the subsequent L4A, L4B and L4C processors.

The different steps of the subsidy application layer preparation are summarized in Figure 1-1 and are detailed in the following sections of this document.

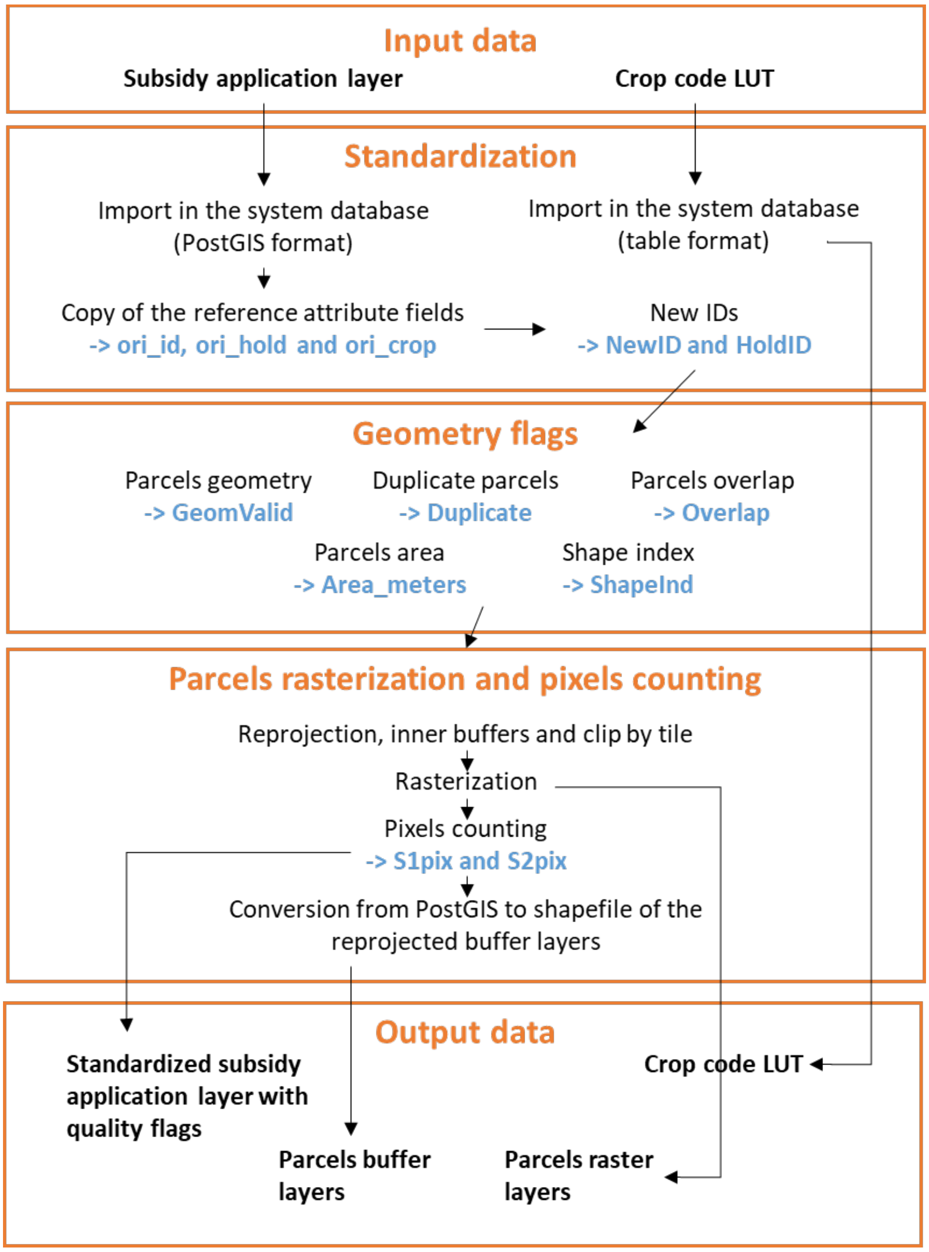

Figure 1-1. Workflow of the subsidy application layer preparation

2. Input data

2.1 Subsidy application layer

The subsidy application layer:

- is a shapefile;
- should be projected, in any projection recognized by the common GDAL functions;
- must at least contain 3 attribute fields which define for each parcel:
    - An unique id of the parcel;
    - A holding id which defines to which holding the parcel belongs: this information is only used for the application of the crop diversification use case after the generation of a crop type product;
    - A crop code which defines the crop type of the parcel.

The names of the attribute fields corresponding to the 3 types of information must be defined when importing the subsidy application layer inside the system.

Table 2-1. Required fields added to the subsidy application layer

Field name	Role	Default value [format]
{any_name}	The initial id of each parcel from the subsidy application layer	[integer or string]
{any_name}	The initial holding id of each parcel from the subsidy application layer	[integer or string]
{any_name}	The initial crop code name associated with each parcel from the subsidy application layer	[integer or string]

2.2 Crop code LUT

The main objective of the crop code LUT is to enable the system to interpret the original crop code defined by the user, in terms of information for the advanced processors. It is used for example to extract the parcels that can be monitored by the advanced processors, as well as for the application of the crop diversification use case after the generation of a crop type product. The crop code LUT is a csv file which contains the information described in Table 2-2.

Table 2-2. Content of the crop code LUT

Field name	Role	Default value [format]
Ori_crop	List of all the possible original crop code from the subsidy application layer	[integer or string]
CTnum	A new sequential number, 1 for each original crop code from the subsidy application layer	[integer]
CT	Crop name associated with the original crop code from the subsidy application layer	[string]

LC	The broad land cover class of each original crop code from the subsidy application layer ○ 0: other natural areas ○ 1: annual crop ○ 2: permanent crop ○ 3: grassland ○ 4: fallow land ○ 5: greenhouse and nursery	[integer]
CTnumL4A	The new crop type code resulting of the grouping of the CTnum for the classification	[integer]
CTL4A	The crop type name associated to CTnumL4A	[string]
CTnumDIV	The crop diversification class code	[integer]
CTDIV	The crop diversification class name	[string]
EAA	Eligible agricultural area: value 1 if the crop type belongs to this category, value 0 otherwise	[integer, binary]
AL	Arable Land: value 1 if the crop type belongs to this category, value 0 otherwise	[integer, binary]
PGrass	Permanent grassland: value 1 if the crop type belongs to this category, value 0 otherwise	[integer, binary]
TGrass	Temporary grassland: value 1 if the crop type belongs to this category, value 0 otherwise	[integer, binary]
Fallow	Fallow land: value 1 if the crop type belongs to this category, value 0 otherwise	[integer, binary]
Cwater	Crop under water: value 1 if the crop type belongs to this category, value 0 otherwise	[integer, binary]

An example of the first lines of a crop code LUT is given in Table 2-3.

Table 2-3. Example of the first lines of a crop code LUT

Ori_crop	CTnum	CT	LC	CTnumL4A	CTL4A	CTnumDIV	CTDIV	EAA	AL	PGrass	TGrass	Fallow	Cwater
174	1	Flower seeds	1	54	Flower_seed	44	Flower_seed	1	1	0	0	0	0
233	2	Wheat winte	1	151	Winter_whea	109	Triticum_win	1	1	0	0	0	0
234	3	Wheat summ	1	142	Triticum_sun	110	Triticum_sun	1	1	0	0	0	0
235	4	Barley winte	1	68	Hordeum_wi	55	Hordeum_wi	1	1	0	0	0	0
236	5	Barley summ	1	69	Hordeum_su	56	Hordeum_su	1	1	0	0	0	0
237	6	Rye (not cut	1	126	Secale	98	Secale	1	1	0	0	0	0
238	7	Oats	1	11	Avena	11	Avena	1	1	0	0	0	0
241	8	chick peas (a	1	37	Chick_peas	91	Pisum	1	1	0	0	0	0
242	9	Beans brown	1	12	Beans	89	Phaseolus	1	1	0	0	0	0
244	10	Peas green /	1	108	Peas	90	Pisum	1	1	0	0	0	0
…	…	…	…	…	…	…	…	…	…	…	…	…	…

3. Detailed workflow

2.1 Standardization

3.1.1 Import of the subsidy applications layer in the system database (conversion in PostGIS)

To facilitate and accelerate the processing, the subsidy application layer is imported as a PostGIS layer in the PostgreSQL database of the system.

3.1.2 Import of the crop code LUT in the system database

To facilitate the integration of the crop code LUT in the different processes, the crop code LUT is imported as a table in the PostgreSQL database of the system.

3.1.3 Copy of the content of the reference attribute fields in ori_id, ori_hold and ori_crop

In this step, the content of the three attribute fields corresponding to the unique parcel id, the holding id and the crop code from the original subsidy application layer is copied in three new attribute fields: ori_id, ori_hold and ori_crop.

The subsidy application layer is updated with the following PostGIS query:

```
update declaration_dataset
set "ori_id" = {attribute field defined by the user}; {string}
set "ori_hold" = {attribute field defined by the user}; {string}
set "ori_crop" = {attribute field defined by the user}; {input format}
```

3.1.4 New IDs

This step adds two **new sequential IDs** to the attribute table of the subsidy application layer:

- NewID (integer): one ID for each parcel, from 1 to the number of parcels;
- HoldID (integer): one ID for each holding, from 1 to the number of holdings.

The parcels and holdings are recognized using the 2 attribute fields defined by the user when importing the subsidy application layer.

The subsidy application layer is updated with the following PostGIS query:

```
update declaration_dataset
set "NewID" = t.new_id,
    "HoldID" = t.hold_id
from (
    select ogc_fid,
           row_number() over (order by ogc_fid) as new_id,
           row_number() over (order by ori_hold) as hold_id
    from declaration_dataset
) t
```

```
where t.ogc_fid = declaration_dataset.ogc_fid;
```

3.1.5 Standardization summary

The summary of the standardization step is given in Table 3-1 and Table 3-2.

Table 3-1. Input and output data of the standardization step

Input data	Description	Default value [format]
{country}_{year}_Decl	Original subsidy application layer	[Shapefile]
Output data	**Description**	**Default value [format]**
{country}_{year}_DeclSTD	The standardized version of the subsidy application layer	[PostGIS]

In the {country}_{year}_DeclSTD file, the following fields have been added to the original subsidy application layer attribute table (Table 3-2).

Table 3-2. New fields added to the subsidy application layer during the standardization step

Field name	Role	Default value [format]
ori_id	Copy of the content of the attribute field defined by the user with the parcel id	[string]
ori_hold	Copy of the content of the attribute field defined by the user with the holding id	[string]
ori_crop	Copy of the content of the attribute field defined by the user with the crop code	[input format: string or integer]
NewID	New sequential ID of the parcel	[integer]
HoldID	New sequential ID of the holdings	[integer]

3.2 Geometry flags

The full definition of each geometry flag is provided here below, as well as the pseudo-code to calculate them.

3.2.1 Parcels geometry -> GeomValid

This flag identifies the parcels:

- for which no polygon exists in the subsidy application layer;
- that have a geometry which is not valid, like self-intersecting polygons.

This flag is binary:

- Value 1 if the parcel has a geometry and the geometry is valid;
- Value 0 if the parcel has no geometry or the geometry is not valid.

The subsidy application layer is updated with the following PostGIS query:

```
update declaration_dataset
set "GeomValid" = ST_IsValid(wkb_geometry);
```

3.2.2 Duplicate parcels -> Duplic

This flag identifies parcels that have the exact same geometry as another.

This flag is binary:

- Value 1 if the parcel has the exact same geometry as another;
- Value 0 if the parcel does not have the exact same geometry as another.

The subsidy application layer is updated with the following PostGIS query:

```
update declaration_dataset
set "Duplic" = "NewID" in (
    select "NewID"
    from (
        select "NewID",
               count(*) over(partition by wkb_geometry) as count
        from lpis
    ) t where count > 1
);
```

3.2.3 Parcels area -> Area_meters

This flag gives the parcel area computed based on the subsidy application layer projected in the national projection. This area does not replace the official declared area which is often stored as another attribute in the original subsidy application layer table.

It is calculated in m² and is stored as an integer value.

The subsidy application layer layer is updated with the following PostGIS query:

```
update declaration_dataset
set "Area_meters" = ST_Area(wkb_geometry);
```

3.2.4 Parcels overlap -> Overlap

This flag identifies the parcels which overlap neighboring parcels with more than 10 % of their area.

This flag is binary:

- Value 1 if the parcel overlaps with other parcels with more than 10% of its area;
- Value 0 if there is no the case.

The subsidy application layer is updated with the following PostGIS query:

```
update declaration_dataset
set "Overlap" = true
where "GeomValid"
and exists (
    select 1
```

```
    from {} lpis
    where t."NewID" != {}."NewID"
    and t."GeomValid"
    and ST_Intersects(t.wkb_geometry, lpis.wkb_geometry)
    having sum(ST_Area(ST_Intersection(t.wkb_geometry, {}.wkb_geometry))) / nullif({}."Area_meters", 0)
> 0.1
);
```

3.2.5 Shape index -> ShapeInd

This flag characterizes the compactness of the parcels. It is calculated with the following equation (Equation 3-1).

$$shapeindex = \frac{Perimeter}{2 * \sqrt{\pi * Area_{UTM}}}$$

Equation 3-1 Calculation of the parcel Shape Index

This flag is stored as a float value. The Shape Index value equals to 1 when the patch is circular and increases without limit as patch becomes more irregular.

The subsidy application layer is updated with the following PostGIS query:

```
update declaration_dataset
set "ShapeInd" = ST_Perimeter(wkb_geometry) / (2 * sqrt(pi() * nullif(ST_Area(wkb_geometry), 0)));
```

3.2.6 Geometry flags summary

The input and output data of the generation of the geometry flags is given in Table 3-3 and Table 3-4.

Table 3-3. Input and output data of the geometry flags step

Input data	Description	Default value [format]
{country}_{year}_DeclSTD	The standardized version of the subsidy application layer	[PostGIS]
Output data	**Description**	**Default value [format]**
{country}_{year}_DeclSTD_geom	The standardized version of the subsidy application layer with geometry flags	[PostGIS]

In {country}_{year}_DeclSTD_geom, the following fields have been added compared to {country}_{year}_DeclSTD (Table 3-4).

Table 3-4. New fields added to the subsidy application layer during the generation of the geometry flags

Field name	Role	Default value [format]
GeomValid	Identify parcels for which no polygon exists in the subsidy application layer or with a not valid geometry	[integer, binary]

Duplic	Identify parcels that have the exact same geometry as another	[integer, binary]
Overlap	Identify parcels which overlaps with neighbouring parcels	[integer, binary]
Area_meters	Parcel area in national projection (m²)	[integer]
ShapeInd	Shape index of the parcel	[float]

3.3 Parcels rasterization and pixels counting

During this step, the parcels are rasterized to facilitate the extraction of the spectral values from the S2 and S1 data performed by the subsequent L4A, L4B and L4C processors. Based on this rasterization, the number of S2 and S1 pixels that are used by parcel for this extraction is also calculated and is added as two additional quality flags to the standardized subsidy application layer with geometry flags. The process also generates parcels buffer layers that can also be used by the subsequent L4A, L4B and L4C processors.

The approach adopted in Sen4CAP to select by parcel the pixels that are used for the extraction of the spectral values of the S2 and S1 data respond to two objectives:

- To keep as much as possible only pixels that cover the parcel and not the surrounding;
- To maximize as much as possible the number of pixels that are used by parcel to extract the spectral values of the S2 and S1 data.

The solution developed is:

- first, to apply an inner buffer to the subsidy application layer with a distance corresponding to half of the resolution of a S2 and S1 pixel;
- second, to select only the S2 and S1 pixels that have their centroid inside this buffer (Figure 3-1).

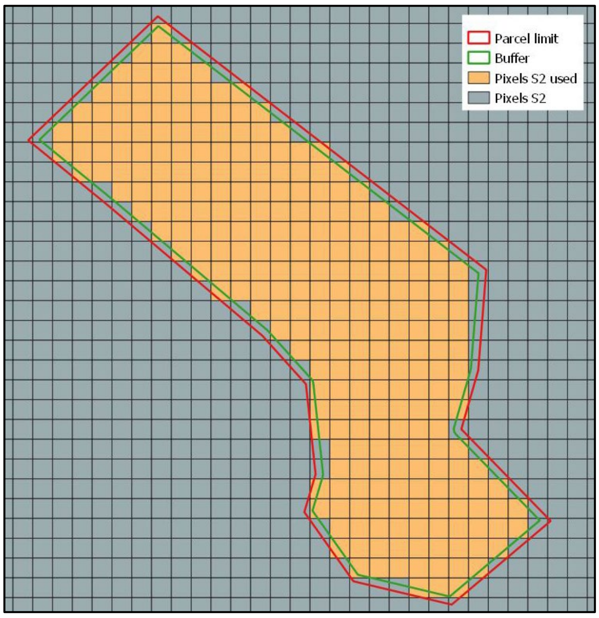

Figure 3-1. S2 pixels selection approach for the spectral values extraction by parcel

It is done in 5 steps:

1. Reprojection: the standardized subsidy application layer is reprojected in the one or several UTM zone(s) that cross the country.
2. Inner buffers: two inner buffer layers are created for each of the reprojected standardized subsidy application layer, at 5 and 10m.
3. Clip by tile: the buffer layers are clipped by tile, using the UTM zone projection corresponding to the tile.
4. Rasterization: the products of the last steps are rasterized using the NewID as the value.
5. Pixels number flags -> S2pix and S1pix: the number of S1 and S2 pixels is calculated for each parcel.

In the implementation, the first three steps are merged into a single PostGIS query. These three steps are thus described together in the following section.

3.3.1 Reprojection, inner buffers and clip by tile

- **Reprojection**

In order to extract the spectral information from the Sentinel data, the standardized subsidy application layer is reprojected in the same projection as the Sentinel data. Depending on their location (in the different UTM zones), the Sentinel data are projected in the corresponding WGS 84 / UTM zone {x} projection. The standardized subsidy application layer is thus reprojected in the one or several UTM zone projections that cross the monitoring region.

- **Inner buffers**

A 5 meters and 10 meters inner buffers are applied the reprojected standardized subsidy application layer(s).

- **Clip by tile**

The multiple reprojected buffer layers are clipped in multiple parts corresponding to each S2 tile extent (bounding-box). Depending on the UTM zone in which the S2 tile is located, the reprojected buffer layer with the corresponding projection is used. The S2 tiles are identified using their reference name made of two digits and three letters (ex. 31UFS).

These three steps are summarized in one single PostGIS query. This query is applied for each of the two buffer values (-5m and -10m) and for each S2 tile of the country:

```
with transformed as (
    select  epsg_code,  ST_Transform(shape_tiles_s2.geom,  Find_SRID('public',  'declaration_dataset',
'wkb_geometry')) as geom
    from shape_tiles_s2
    where tile_id = {}
)
select "NewID", ST_Buffer(ST_Transform(wkb_geometry, epsg_code), {})
from lpis, transformed
where ST_Intersects(wkb_geometry, transformed.geom);
```

- **Summary**

The input and output data of these three steps are detailed in Table 3-5.

Table 3-5. Input and output data of the reprojection, inner buffers and clip by tile steps

Input data	Role	Default value [format]
{country}_{year}_DeclSTD_geom	The standardized version of the subsidy application layer with geometry flags	[PostGIS]
extent_tile_{i}	Shapefile with the bounding box of each S2 tile; {i} = name of the tile (ex. 31UFS)	[shapefile] (nr of tiles)
Output data	**Role**	**Default value [format]**
{country}_{year}_DeclSTD_geom_buffer_5m_{i}	{country}_{year}_DeclSTD_geom_buffer_5m_{x} clipped to the extent of the each S2 tile; {i} = name of the tile (ex. 31UFS)	[PostGIS] (nr of tiles)
{country}_{year}_DeclSTD_geom_buffer_10m_{i}	{country}_{year}_DeclSTD_geom_buffer_10m_{x} clipped to the extent of the each S2 tile; {i} = name of the tile (ex. 31UFS)	[PostGIS] (nr of tiles)

3.3.2 Rasterization

The reprojected and clipped by S2 tile buffer layers are rasterized using the NewID field as the value to obtain a one-band raster with each pixel having the value of the NewID of the parcels. The tile extents are used as reference frame. The NewID value of the parcel is assigned to a pixel only if its centroid is located inside of the corresponding reprojected and clipped by S2 tile buffer layer.

Rasterization of {country}_{year}_DeclSTD_geom_buffer_5m_{i}:

- value = NewID;
- resolution = 10 m;
- tile extent used as reference frame;
- rule = centroid of the pixel inside the inner buffer.

Rasterization of {country}_{year}_DeclSTD_geom_buffer_10m_{x}_{i}:

- value = NewID;
- resolution = 20 m;
- tile extent used as reference frame;
- rule = centroid of the pixel inside the inner buffer.

The rasterization is done using `gdal_rasterize`, as follows:

```
gdal_rasterize -q \
    -a NewID \
    -a_srs EPSG_CODE \
    -te XMIN YMIN XMAX YMAX \
    -tr RES RES\
    -sql SQL \
    -ot Int32 \
    -co COMPRESS=DEFLATE \
    -co PREDICTOR=2 \
```

```
INPUT OUTPUT
```

The input and output data of this step are detailed in Table 3-6.

Table 3-6. Input and output data of the rasterization step

Input data	Role	Default value [format]
{country}_{year}_DeclSTD_geom_buffer_5m_{i}	{country}_{year}_DeclSTD_geom_buffer_5m_{x} clipped to the extent of the each S2 tile; {i} = name of the tile (ex. 31UFS)	[PostGIS] (nr of tiles)
{country}_{year}_DeclSTD_geom_buffer_10m_{i}	{country}_{year}_DeclSTD_geom_buffer_10m_{x} clipped to the extent of the each S2 tile; {i} = name of the tile (ex. 31UFS)	[PostGIS] (nr of tiles)
Output data	**Role**	**Default value [format]**
{country}_{year}_decl_{i}_S2	Raster of all the used S2 pixels by parcel (value as NewID); resolution = 10 m; {i} = name of the tile (ex. 31UFS)	[tif] (nr of tiles)
{country}_{year}_decl_{i}_S1	Raster of all the used S1 pixels by parcel (value as NewID); resolution = 20 m; {i} = name of the tile (ex. 31UFS)	[tif] (nr of tiles)

3.3.3 Pixels counting -> S1pix and S2pix

The number of S2 and S1 pixels that will be used to extract spectral values from the S2 and S1 data is calculated by parcel. This information is added to the standardized subsidy application layer with geometry flags (Table 3-7 and Table 3-8).

It is performed following the pseudo-code:

```
for each buffer in [5, 10]:
    for each tile in the S2 tiles:
        compute the histogram of the rasterization output
        sort the histogram by NewID
    merge histograms, summing the pixel counts across tiles
    update the PostGIS table with the resulting counts
```

The input and output data of this step are detailed in Table 3-7.

Table 3-7. Input and output data of the pixels number flags step

Input data	Role	Default value [format]
{country}_{year}_decl_{i}_S2	Raster of all the used S2 pixels by parcel (value as NewID); resolution = 10 m; {i} = name of the tile (ex. 31UFS)	[tif] (nr of tiles)

{country}_{year}_decl_{i}_S1	Raster of all the used S1 pixels by parcel (value as NewID); resolution = 20 m; {i} = name of the tile (ex. 31UFS)	[tif] (nr of tiles)
{country}_{year}_DeclSTD_geom	The standardized version of the subsidy application layer with geometry flags	[PostGIS]
Output data	**Role**	**Default value [format]**
{country}_{year}_DeclSTD_quality_flags	The standardized version of the subsidy application layer with the quality flags: geometry and pixels number flags	[PostGIS]

In {country}_{year}_DeclSTD_quality_flags, the following fields have been added compared to {country}_{year}_DeclSTD_geom (Table 3-8).

Table 3-8. New fields added to the subsidy application layer during the pixels number flags step

Field name	Role	Default value [format]
S1pix	Indicates the number of used S1 pixels in the parcel	[integer]
S2pix	Indicates the number of used S2 pixels in the parcel	[integer]

3.3.4 Conversion from PostGIS to shapefile of the reprojected buffer layers

This step aims at converting the buffer layers which are in PostGIS format in shapefile format (Table 3-9).

```
ogr2ogr -sql ‘select "NewID", ST_Buffer(ST_Transform(wkb_geometry, EPSG_CODE), BUFFER)’ OUTPUT INPUT
```

Table 3-9. Input and output data of the conversion from PostGIS to shapefile of the reprojected buffer layers step

Input data	Description	Default value [format]
{country}_{year}_DeclSTD_geom_buffer_5m_{x}	DeclSTD _geom_buffer_5m reprojected; {x} = EPSG reference of the projection (ex. 32631)	[PostGIS] (number of UTM zone(s))
{country}_{year}_DeclSTD_geom_buffer_10m_{x}	DeclSTD_geom_buffer_10m reprojected; {x} = EPSG reference of the projection (ex. 32631)	[PostGIS] (number of UTM zone(s))
Output data	**Description**	**Default value [format]**
{country}_{year}_DeclSTD_geom_buffer_5m_{x}	DeclSTD _geom_buffer_5m reprojected; {x} = EPSG reference of the projection (ex. 32631)	[shapefile] (number of UTM zone(s))

{country}_{year}_DeclSTD_geom_buffer_10m_{x}	DeclSTD_geom_buffer_10m reprojected; {x} = EPSG reference of the projection (ex. 32631)	[shapefile] (number of UTM zone(s))

4. Ouput data

4.1 Standardized subsidy application layer with quality flags

The first output is a PostGIS layer in the PostgreSQL database of the system: the standardized subsidy application layer with the quality flags (Table 4-1). It:

- is a PostGIS layer;
- is projected in national projection;
- has the following name: **decl_{site_name}_{year}**;
- has the same number of rows (parcels) than the original subsidy application layer.

Table 4-1. Output 1: standardized subsidy application layer with quality flags

Output data	Description	Default value [format]
decl_{site_name}_{year}	The standardized version of the subsidy application layer with the quality flags: geometry and spectral information	[PostGIS]

It contains the following attribute fields (fields in orange are already present in the original subsidy application layer) (Table 4-2).

Table 4-2. Attribute fields in the standardized subsidy application layer with quality flags

Field name	Role	Default value [format]
Ori attributes	All the original attributes of the original delaration dataset	[integer, float or string]
ori_id	Copy of the content of the attribute field defined by the user with the parcel id	[string]
ori_hold	Copy of the content of the attribute field defined by the user with the holding id	[string]
ori_crop	Copy of the content of the attribute field defined by the user with the crop code	[input format: string or integer]
NewID	New sequential ID of the parcel	[integer]
HoldID	New sequential ID of the holdings	[integer]
GeomValid	Identify parcels for which no polygon exists in the subsidy application layer or with a not valid geometry	[integer, binary]
Duplic	Identify parcels that have the exact same geometry as another	[integer, binary]
Area_meters	Parcel area in the UTM projection (m^2)	[integer]
Overlap	Identify parcels which overlaps with neighbouring parcels	[integer, binary]
ShapeInd	The crop type name	[float]

S1pix	Indicates the number of used S1 pixels in the parcel	[integer]
S2pix	Indicates the number of used S2 pixels in the parcel	[integer]

4.2 Crop code LUT

The crop code LUT is imported as a table in the PostgreSQL database of the system (Table 4-3). It:

- is a table;
- has the following name: **lut_{site_name}_{year}.**

Table 4-3. Output 2: crop code LUT in the PostgreSQL database of the sytem

Output data	Description	Default value [format]
lut_{site_name}_{year}	The crop code LUT	[table]

It contains the following attribute fields (Table 4-4).

Table 4-4. Attribute fields in the crop code LUT

Field name	Role	Default value [format]
Ori_crop	List of all the possible original crop code from the subsidy application layer	[integer or string]
CTnum	A new sequential number, 1 for each original crop code from the subsidy application layer	[integer]
CT	Crop name associated with the original crop code from the subsidy application layer	[string]
LC	The broad land cover class of each original crop code from the subsidy application layer ○ 0: other natural areas ○ 1: annual crop ○ 2: permanent crop ○ 3: grassland ○ 4: fallow land ○ 5: greenhouse and nursery	[integer]
CTnumL4A	The new crop type code resulting of the grouping of the CTnum for the classification	[integer]
CTL4A	The crop type name associated to CTnumL4A	[string]
CTnumDIV	The crop diversification class code	[integer]
CTDIV	The crop diversification class name	[string]
EAA	Eligible agricultural area: value 1 if the crop type belongs to this category, value 0 otherwise	[integer, binary]
AL	Arable Land: value 1 if the crop type belongs to this category, value 0 otherwise	[integer, binary]

PGrass	Permanent grassland: value 1 if the crop type belongs to this category, value 0 otherwise	[integer, binary]
TGrass	Temporary grassland: value 1 if the crop type belongs to this category, value 0 otherwise	[integer, binary]
Fallow	Fallow land: value 1 if the crop type belongs to this category, value 0 otherwise	[integer, binary]
Cwater	Crop under water: value 1 if the crop type belongs to this category, value 0 otherwise	[integer, binary]

4.3 Parcels buffer layers

The second outputs are the parcels reprojected buffer layers (5m and 10m inner buffers). These buffers are reprojected in the WGS 84 / UTM zone {x} projections that correspond to all the UTM zones that cross the country. It only contain one field, the NewID of the parcel.

The parcels buffer layers:

- are shapefiles;
- are projected in WGS 84 / UTM zone {x} (as many as the number of UTM zones that cross the country);
- do not have necessarily the same number of lines (parcels) than the original subsidy application layer.

Table 4-5. Output 3: parcels buffer layers

Output variable	Description	Default value [format]
{country}_{year}_DeclSTD_geom_buffer_5m_{x}	DeclSTD _geom_buffer_5m reprojected; {x} = EPSG reference of the projection (ex. 32631)	[shapefile] (number of UTM zone(s))
{country}_{year}_DeclSTD_geom_buffer_10m_{x}	DeclSTD_geom_buffer_10m reprojected; {x} = EPSG reference of the projection (ex. 32631)	[shapefile] (number of UTM zone(s))

The parcels buffer layers only contain one field, the NewID of the parcel (Table 4-6).

Table 4-6. Attribute field of the parcels buffer layers

Field name	Role	Default value [format]
NewID	New sequential ID of the parcel	[integer]

4.4 Parcels raster layers

The third outputs are the rasters that are produced for both data (S2 and S1) and by tile, with the NewID as value. Only the pixels that have their centroid located in the corresponding buffer layers (5m for S2 and 10m for S1) have been assigned the NewID value of the parcel.

These layers:

- are .tif files;

- are produced by S2 tile;
- are projected in the WGS 84 / UTM zone {x} corresponding to the UTM zone of the S2 tile;
- value = NewID of the parcels.

Table 4-7. Ouput 4: parcels raster layers

Output variable	Role	Default value [format]
{country}_{year}_decl_{i}_S2	Raster of all the used S2 pixels by parcel (value as NewID); resolution = 10 m; {i} = name of the tile (ex. 31UFS)	[tif] (nr of tiles)
{country}_{year}_decl_{i}_S1	Raster of all the used S1 pixels by parcel (value as NewID); resolution = 20 m; {i} = name of the tile (ex. 31UFS)	[tif] (nr of tiles)