Quick Start Guide — Living Earth LCCS

2026-04-29

Table of contents

1 What is Living Earth LCCS? 1

2 Before You Begin 2

3 Core Concepts 3
3.1 The LCCS Classification Hierarchy . . . 3
3.2 Input Data . . . 3

4 Step-by-Step Workflow 4
4.1 Step 1 — Prepare Your Environmental Descriptor Inputs . . . 4
4.2 Step 2 — Import the Package . . . 4
4.3 Step 3 — Load the L3 Classification Layers . . . 5
4.4 Step 4 — Generate L4 Descriptors . . . 5
4.5 Step 5 — Write Output to File . . . 6

5 Using an OpenDataCube Backend 6

6 Using Plugins 7

7 Understanding the Output 7

8 Next Steps 8

9 Key References 8

1 What is Living Earth LCCS?

Living Earth LCCS is an open-source Python package that generates land cover maps from Earth Observation (EO) data using the **Food and Agriculture Organisation (FAO) Land**

Cover Classification System (LCCS v2). It has been applied at national scales in Australia, Wales, Papua New Guinea, and Switzerland, and is designed to be globally scalable.

The classification approach works by combining **Environmental Descriptors (EDs)** — measurable properties of the landscape retrieved from satellite data — through a hierarchical decision-tree framework to produce standardised land cover classes. These classes fall into three levels:

Descriptor type	Role
Overarching EDs (OEDs)	Determine the 8 base-level LCCS Level 3 classes (e.g. vegetated vs. bare, terrestrial vs. aquatic, managed vs. natural)
Essential EDs (EEDs)	Add detail within each L3 class (e.g. lifeform, water hydroperiod) to yield Level 4 classifications
Additional EDs (AEDs)	Provide further biophysical context (e.g. soil acidity, dominant species, urban density) — not part of the core FAO taxonomy

The resulting maps can represent over 12,000 unique land cover classes, each carrying biophysical meaning and expressed as a numeric code for use in GIS applications.

---

2 Before You Begin

Make sure you have completed installation as described in the Installation Guide. In summary, you need:

- Python 3.8
- The `livingearth_lccs` package installed
- GDAL available in your environment
- (Optional) An OpenDataCube (ODC) instance if using cube-based data access

Verify your installation:

```
import livingearth_lccs
print(livingearth_lccs.__version__)
```

---

3 Core Concepts

3.1 The LCCS Classification Hierarchy

The FAO LCCS operates in two stages:

Stage 1 — Dichotomous phase (Levels 1–3)

Three binary decisions produce 8 possible Level 3 (L3) classes:

```
Level 1:  Vegetation (A)     vs.  Bare (B)
Level 2:  Terrestrial (1)    vs.  Aquatic (2)
Level 3:  Managed (odd)      vs.  Natural (even)
```

Valid L3 codes are: `A11`, `A12`, `A23`, `A24`, `B15`, `B16`, `B27`, `B28`.

Stage 2 — Modular-hierarchical phase (Level 4)

Each L3 class is further described by approximately a dozen categorical L4 fields (41 in total). These encode attributes such as lifeform, canopy height, crop type, and water seasonality. L4 classes operate hierarchically: some fields must be populated before others become meaningful.

3.2 Input Data

Living Earth accepts two types of data sources:

- **GDAL-compatible rasters** — any local or remote file format supported by GDAL (GeoTIFF, NetCDF, HDF, COG, etc.)
- **OpenDataCube products** — data indexed in an ODC instance (e.g. Digital Earth Australia, the Welsh Data Cube)

Inputs correspond to Environmental Descriptors and must be provided as classified or continuous raster layers (e.g. fractional vegetation cover as a percentage, water hydroperiod in months per year, lifeform as a categorical code).

---

4 Step-by-Step Workflow

4.1 Step 1 — Prepare Your Environmental Descriptor Inputs

Each input raster should correspond to one Environmental Descriptor. Ensure your data:

- Is spatially aligned (same CRS, extent, and resolution)
- Uses the correct units or category codes as expected by the LCCS layer definitions
- Covers your area of interest

For example, a minimal set of OEDs to produce L3 classifications might include:

Descriptor	Format	Example values
Presence of vegetation cover	Binary raster	0 = bare, 1 = vegetated
Terrestrial / aquatic	Binary raster	0 = aquatic, 1 = terrestrial
Managed / natural	Binary raster	0 = natural, 1 = managed

To generate L4 detail, add EEDs such as:

Descriptor	Format	Example values
Lifeform (vegetation)	Categorical raster	e.g. 1 = woody, 2 = herbaceous
Annual water hydroperiod	Continuous raster (months/yr)	0–12
Canopy height	Continuous raster (metres)	0–50+

4.2 Step 2 — Import the Package

```
import livingearth_lccs
```

For GDAL-based raster workflows, you will typically also need:

```
import numpy as np
from osgeo import gdal
```

4.3 Step 3 — Load the L3 Classification Layers

The dichotomous L3 phase is handled by the Level 3 classification module. Pass your OED arrays (as NumPy arrays or equivalent) into the L3 decision tree:

```
from livingearth_lccs import l3_layers_lccs

# Example: load your OED rasters as numpy arrays
# (replace these with your actual data loading steps)
vegetation = gdal.Open("vegetation_cover.tif").ReadAsArray()
terrestrial = gdal.Open("terrestrial_aquatic.tif").ReadAsArray()
managed = gdal.Open("managed_natural.tif").ReadAsArray()

# Generate L3 classification
l3_result = l3_layers_lccs.classify(
    vegetation=vegetation,
    terrestrial=terrestrial,
    managed=managed
)
```

> **Note**
>
> Exact function signatures and argument names should be confirmed against the module docstrings in the repository (`l3_layers_lccs.py`) or the API reference in the documentation.

4.4 Step 4 — Generate L4 Descriptors

Once L3 classes are available, pass the EED layers into the L4 module to produce the full modular-hierarchical classification. The L4 code is documented with docstrings in `l4_layers_lccs.py`:

```
from livingearth_lccs import l4_layers_lccs

lifeform = gdal.Open("lifeform.tif").ReadAsArray()
hydroperiod = gdal.Open("hydroperiod.tif").ReadAsArray()

l4_result = l4_layers_lccs.classify(
    l3_classification=l3_result,
    lifeform=lifeform,
    hydroperiod=hydroperiod,
```

```
    # ... add further EEDs as available
)
```

4.5 Step 5 — Write Output to File

The classification result is a raster array of integer codes (each uniquely identifying a land cover class). Write it out using GDAL:

```
driver = gdal.GetDriverByName("GTiff")
out_ds = driver.Create(
    "landcover_output.tif",
    l4_result.shape[1],   # columns
    l4_result.shape[0],   # rows
    1,                    # number of bands
    gdal.GDT_Int32
)

# Copy georeferencing from one of your input files
ref_ds = gdal.Open("vegetation_cover.tif")
out_ds.SetGeoTransform(ref_ds.GetGeoTransform())
out_ds.SetProjection(ref_ds.GetProjection())

out_ds.GetRasterBand(1).WriteArray(l4_result)
out_ds.FlushCache()
out_ds = None
print("Output written to landcover_output.tif")
```

---

5 Using an OpenDataCube Backend

If your data is indexed in an OpenDataCube (ODC) instance, products can be loaded directly through the ODC API before passing them to the LCCS classification routines.

```
import datacube

dc = datacube.Datacube(app="livingearth_quickstart")

# Load a product (example: fractional vegetation cover)
```

```
data = dc.load(
    product="fc_percentile_albers_annual",
    x=(149.0, 150.0),
    y=(-36.0, -35.0),
    time=("2020-01-01", "2020-12-31"),
    output_crs="EPSG:3577",
    resolution=(-30, 30)
)

fvc = data["PV"].values  # Photosynthetic Vegetation fraction
```

Pass the resulting NumPy arrays into the LCCS classification modules as shown in Steps 3 and 4 above.

---

6 Using Plugins

Living Earth supports a plugin system for extending its input handling and for generating EDs from Analysis Ready Data (ARD). Plugins can be used to:

- Compute Environmental Descriptors directly from ARD (e.g. deriving hydroperiod from Sentinel-1 backscatter time series)
- Integrate new data sources or sensors
- Add virtual products

Consult the plugin documentation and the `plugins/` directory in the repository for examples of how to develop and register a plugin.

---

7 Understanding the Output

Each pixel in the output raster contains a unique integer code representing a full LCCS classification. For example:

- An integer corresponding to `A11` (vegetated, terrestrial, managed) combined with L4 attributes such as *herbaceous lifeform, annual crop* encodes a specific agricultural land cover class.

- The full lookup table of all 573,307+ unique codes and their descriptions is available in the supplementary material of the Living Earth publication (Lewis et al., 2021, *Big Earth Data*).

To interpret codes, refer to the `l4_layers_lccs.py` docstrings or the LCCS documentation, which describe each field, its permitted values, and its dependencies.

---

8 Next Steps

Once you have produced your first land cover map, you can:

- **Compare maps across time** to detect land cover change using the Living Earth Global Change Taxonomy
- **Generate habitat maps** through direct translation of selected LCCS classes (as done for Wales)
- **Quantify uncertainty** associated with each Environmental Descriptor and propagate it through the classification
- **Explore the interactive map** of existing Living Earth products at livingearthhub.org

---

9 Key References

- **Source code:** https://bitbucket.org/au-eoed/livingearth_lccs
- **Documentation:** https://livingearth-lccs.readthedocs.io/en/latest/
- **Living Earth Hub:** http://livingearthhub.org
- Lewis et al. (2021). *Living Earth: Implementing national standardised land cover classification systems for Earth Observation in support of sustainable development.* Big Earth Data, 5(3). https://doi.org/10.1080/20964471.2021.1948179

> **!** Important
>
> The official documentation at `livingearth-lccs.readthedocs.io` contains the most up-to-date API reference, function signatures, and worked examples. Always consult it alongside this guide, as specific argument names and module structures may evolve between versions.