University of Tasmania
GPO Box 367
Hobart 7001
Australia

imos.org.au

May 2026

IMOS NESP 5.9

Seagrass Aggregated Data Product

Product Technical Document

Version 1.1

Thomas Galindo
Australian Ocean Data Network AODN / IMOS

Australia's Integrated Marine Observing System (IMOS) is enabled by the National Collaborative Research Infrastructure Strategy (NCRIS). It is operated by a consortium of institutions as an unincorporated joint venture, with the University of Tasmania as Lead Agent.

IMOS acknowledges the Traditional Custodians and Elders of the land and sea on which we work and observe, and recognise them as Australia's first marine scientists and carers of sea Country. We pay our respects to Aboriginal and Torres Strait Islander peoples past and present.

1. Version History

Version	Date	Comments	Author
v1.0	December-2025	Initial Release	Galindo, T
v1.1	May-2026	Minor Revision	Galindo, T

Citation

Galindo, T. (2026). Seagrass Aggregated Data Product Technical Document. Version 1.1 Integrated Marine Observing System.

Copyright/Creative Commons Licence

Table of Contents

2. Background

2.1 Overview

Basis of the Seagrass Aggregated Data Product

The National Seagrass Data Aggregation consolidates many heterogeneous seagrass datasets into a single, standardised data product developed under Project 5.9. The purpose of the aggregation is to provide an analysis ready, nationally consistent seagrass dataset that supports ecological assessment, long term monitoring, and habitat modelling.

Source datasets were contributed by multiple Australian research institutions and government agencies, including IMAS, TropWATER (James Cook University), CSIRO, AIMS, and state and territory environment agencies. Data were accessed primarily via Seamap Australia GeoServer services, with supplementary static exports from institutional repositories.

The aggregation integrates data derived from a wide range of survey methods, including field-based GPS mapping, aerial and boat-based surveys, underwater video and sonar, UAV orthomosaics, and satellite-derived products. As a result, source datasets vary substantially in spatial resolution, temporal precision, taxonomic encoding, and data structure.

Spatial coverage spans Australian coastal waters nationwide, with dense coverage in regions including the Great Barrier Reef, Torres Strait, Gulf of Carpentaria, Moreton Bay, Hervey Bay, Cockburn Sound, and temperate southern coastlines. Temporal coverage extends from 1967 to 2024.

The product records presence and absence of 13 seagrass taxa. Species names are mapped to the Codes for Australian Aquatic Biota (CAAB) and aligned with World Register of Marine Species (WoRMS) taxonomy. Unresolved taxa (e.g. sp. or spp.) are standardised to the nearest resolved parent taxon, and biomass observations are retained where available.

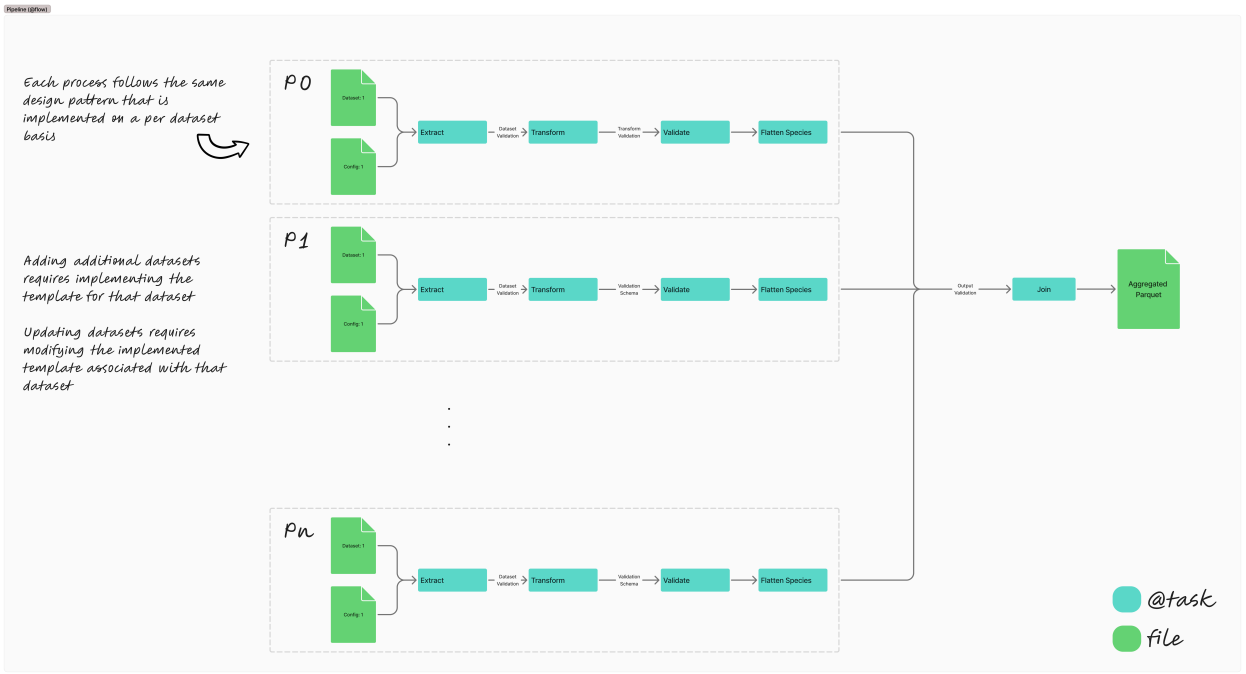
The final output is delivered as a long format Parquet dataset, with one record per species observation per site visit. Spatial enrichment includes H3 indices (resolution 15) and Australian Marine Region tags. Use of columnar storage and compression reduces the aggregated dataset from over 2 GB of source exports to approximately 230 MB.

3. Methodology

The Seagrass Aggregated Data Product (Data Product) relies on the standard Extract, Transform, and Load (ETL) methodology.

The ETL process to generate the Seagrass Aggregated Data Product is two step:

1. Transform source data to a harmonised intermediary standardised format
2. Transform intermediary harmonised data into singular data product



3.1 Extract

The Seagrass Aggregated Data Product is comprised of the datasets documented here.

Data Source

The underlying data is typically sourced from geo-servers maintained by:

1. IMAS
2. JCU
3. NESP

Extraction of the original datasets was conducted prior construction of the ETL pipeline.

There are plans to replace the static exports with dynamic exports from source.

Validation

Each dataset was analysed to understand its scheme and is validated against its own custom schema custom schema.

While there are similarities between each dataset, the column names, biological information and geometries have subtle differences that had to be accounted for per dataset.

Minimum Required Fields

The minimum required fields are the standard biological datasets: when, where and what.

Column	Typical Type	Description	Typical Data Quality
Geometry	string	Usually a WKT encoding of the biological occurrence	Good
Date	string	Usually an ISO8601 or similar encoded string	Poor
Species	string or comma separated string	Either a list of scientific names or one hot encoded scientific names	Good

This is a minimum example of a categorical encoded seagrass dataset schema:

```
"pyarrow_schema": {
    "fields": [

        // When
        {
            "name": "Date",
            "type": "date32[day]",
            "nullable": false,
            "date_encoding": "start_date"
        },

        // Where: The wkt geometry encoded in epsg:4326
        {
            "name": "geom",
            "type": "string",
            "nullable": false,
            "projection": 4326,
            "format": "wkt"
        },

        // What: Species present at the geometry
        {
            "name": "SM_HAB_CLS",
            "type": "string",
            "nullable": false,
            "aggregate_encoding": "presence"
        },
    ],
}
```

At runtime, the input data is validated per dataset against its custom schema, ensuring correct type interpretation for CSV data.

Extraction

Extraction is handled via the `common.file.S3Block` class in conjunction with the `common.factory.PyarrowTableFactory` class.

These serializable pydantic classes allow the extraction to live as JSON configuration files, saved in this project as `<dataset>.factory.json` files and ensure type valided arrow is generated from CSV inputs:

```
{
    "file_interface": {
        "block_name": "processing-bucket",
        "path": "seagrass/TS_JCU_Western-Torres-Strait_2020_Seagrass-community-
type.csv"
    },
    "format": "csv",
    "pyarrow_schema": {
        "fields": [

            // When
            {
                "name": "Date",
                "type": "date32[day]",
                "nullable": false,
                "date_encoding": "start_date"
            },

            // Where: The wkt geometry encoded in epsg:4326
            {
                "name": "geom",
                "type": "string",
                "nullable": false,
                "projection": 4326,
                "format": "wkt"
            },

            // What: Species present at the geometry
            {
                "name": "SM_HAB_CLS",
                "type": "string",
                "nullable": false,
                "aggregate_encoding": "presence"
            },
        ],
    }
}
```

3.2 Transform

Because the underlying seagrass data sources have unique time, place and species encodings, the Seagrass Aggregate Data Product transformations are bespoke to each source.

Per Dataset Transformation Pattern

The typical transformation pattern procedure is as follows:

1. Extract geometry
2. Extract date
3. Extract species presence/absence

Extract Geometry

Because the export data uses a WKT export query paramter, the geometry column almost always contains WKT compliant geometry information. This allows the latitude and longitude to be located.

Where the geometry column is unavailable, the latitude and longitude are used as a substitue.

Extract Date

Date information is usually ISO8601 compliant, but in many cases is not.

The dates are manually audited and mapped where the date encoding is non ISO8601 compliant and follows inconsistent patterns, eg:

```
start_year = dates.map(
    {
        "16/12/2020": 2020,
        "Dec-19": 2019,
        "2015-2016": 2015,
        "10,17,19,27/5/22": 2022,
        "December-2010, November-2011": 2010,
        "Jun-21": 2021,
        "May/June-2012": 2012,
        "2-4/11/2020": 2020,
        "May-2020": 2020,
        "14-18/11/22": 2022,
        "October, 2017": 2017,
        "Aug-20": 2020,
        "9/02/2022,16/02/2022": 2022,
        "October-2010": 2010,
        "2005,2008": 2005,
        "27-28/10/2021": 2021,
        "21/07/2022": 2022,
        "22-24/09/2020": 2020,
        "Nov-19": 2019,
        ...
    }
)

start_month = dates.map(
    {
        "16/12/2020": 12,
        "Dec-19": 12,
        "2015-2016": 0,
        "10,17,19,27/5/22": 5,
        "December-2010, November-2011": 12,
        "Jun-21": 6,
        "May/June-2012": 5,
        "2-4/11/2020": 11,
        "May-2020": 5,
        "14-18/11/22": 11,
        "October, 2017": 10,
        "Aug-20": 8,
        "9/02/2022,16/02/2022": 2,
        "October-2010": 10,
        "2005,2008": 1,
        "27-28/10/2021": 10,
        "21/07/2022": 7,
        "22-24/09/2020": 9,
        "Nov-19": 11,
        ...
    }
)

start_day = dates.map(
    {
        "16/12/2020": 16,
        "Dec-19": 0,
        "2015-2016": 0,
        "10,17,19,27/5/22": 10,
```

```
        "December-2010, November-2011": 0,
        "Jun-21": 0,
        "May/June-2012": 0,
        "2-4/11/2020": 2,
        "May-2020": 0,
        "14-18/11/22": 14,
        "October, 2017": 0,
        "Aug-20": 0,
        "9/02/2022,16/02/2022": 9,
        "October-2010": 0,
        "2005,2008": 0,
        "27-28/10/2021": 27,
        "21/07/2022": 21,
        "22-24/09/2020": 22,
        "Nov-19": 0,
    }
)
```

Extract Species Presence/Absence

The typical species data transforms are to:

1. Re-encode delimited string columns
2. Re-encode one hot encoded species
3. Map species to the Codes for Australian Aquatic Biota controlled vocabulary (CAAB)
4. Disambiguation of species

Delimited String Species

The presence/absence of species are commonly encoded in a delimited string column.

In these cases, the column values were audited and mapped to the CAAB controlled vocabulary.

eg:

Species
"HO, HA, AO"
"HO, HA"
"AO"

One Hot Encoded Species

The presence/absence of species are also commonly one hot encoded.

In these cases, the column names were audited and mapped to the CAAB controlled vocabulary.

eg:

HO	HA	AO
TRUE	TRUE	TRUE
TRUE	TRUE	FALSE
FALSE	FALSE	TRUE

Species Mapping

The species were mapped to the CAAB controlled vocabulary. Details of this vocabulary are found in the enumeration.py.

Species Disambiguation

Another issue commonly encountered is the usage of `sp.` or `spp.`. In such instances, the scientific name is rolled up to the nearest parent.

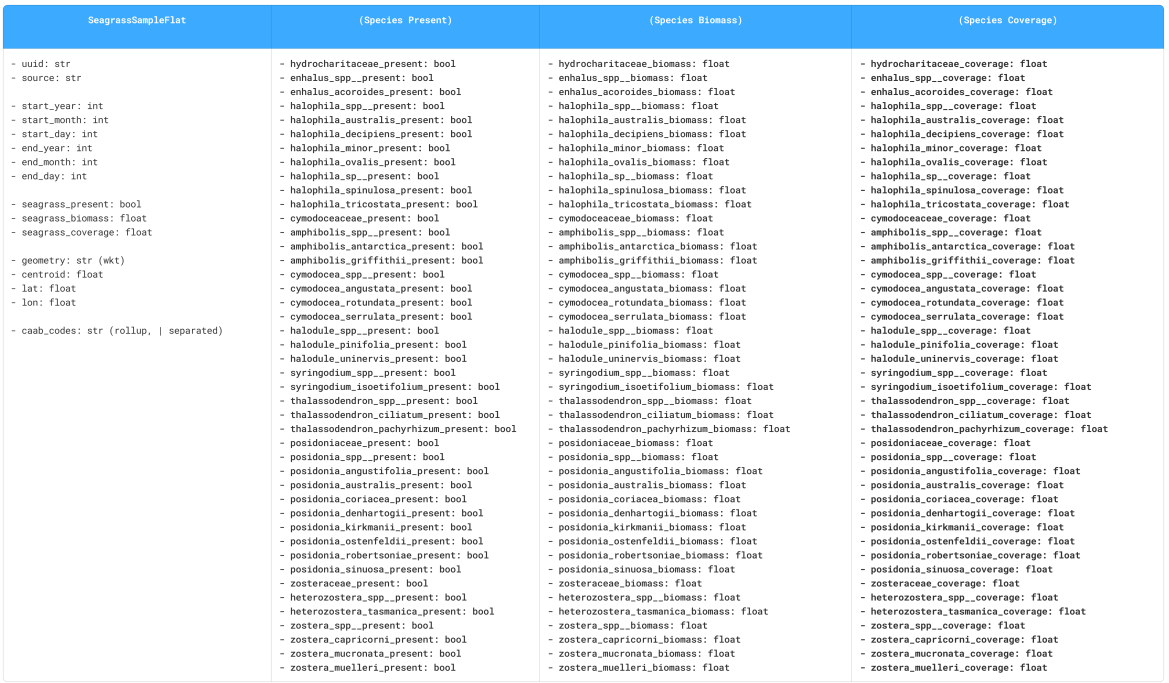
Per Dataset Transformation Validation

Once a dataset it is validated against an intermediary schema:

name	type	nullable
source	string	false
source_id	string	false
metadata_link	string	false
*When *	-	-
start_year	int16	false
start_month	int8	true
start_day	int8	true
end_year	int16	true
end_month	int8	true
end_day	int8	true
*Where *	-	-
geometry	string	false
What	-	-
seagrass_present	bool	false

thalassia_present	bool	true
thalassia_hemprichii_present	bool	true
hydrocharitaceae_present	bool	true
enhalus_present	bool	true
enhalus_acoroides_present	bool	true
halophila_present	bool	true
Species continues...	-	-

The full output schema is shown below:



Post Processing

Once extract transform is complete per dataset, the harmonised data is stacked vertically ready for post processing.

Geometry Pre-processing

Before pivoting to long format, the stacked geometries are pre-processed:

- **Split multi-polygons**: `MULTIPOLYGON` geometries are split into individual `POLYGON` rows so each row represents a single geometry
- **Repair invalid geometries**: invalid geometries are repaired using `shapely.make_valid` to ensure all geometries are topologically valid

- **Extract centroid coordinates**: `decimalLatitude` and `decimalLongitude` are extracted from the centroid of each geometry

Geospatial Enrichment

After geometry pre-processing, each record is enriched with:

- `h3Index` — A hexadecimal string representing an H3 polygon at resolution 5, derived from `decimalLatitude` and `decimalLongitude`
- `australianMarineRegionsTags` — A `|` separated tag column featuring common Australian marine regions

Wide to Long Pivot

The intermediary transformed data is kept in wide format to simplify translation of species.

The final dataset however is pivoted to long format to improve usability.

Sort Order

The final dataset is sorted by `sourceID` then `_temporal_extent` (ascending). Sorting by source dataset first groups all observations from the same dataset contiguously. The secondary sort by date further clusters observations that share similar spatial locations and measurement values, improving dictionary and run-length encoding compression across columns such as `h3Index`, `decimalLatitude`, `decimalLongitude`, and `footprintWKT`.

Output Validation

The long format output table is validated against the schema below (Darwin Core naming convention):

name	type	nullable	description	unit
datasetName	string	false	The name identifying the data set from which the record was derived	
sourceID	string	false	The source dataset row id	
metadataLink	string	false	The source dataset metadata link	

startYear	int16	false	The start year of the measurement	
startMonth	int8	true	The start month of the measurement	
startDay	int8	true	The start day of the measurement	
endYear	int16	true	The end year of the measurement (if applicable)	
endMonth	int8	true	The end month of the measurement (if applicable)	
endDay	int8	true	The end day of the measurement (if applicable)	
footprintWKT	string	false	The WKT encoded geometry of the measurement	
decimalLatitude	float32	false	The latitude of the measurement, taken from the centroid of the geometry	
decimalLongitude	float32	false	The longitude of the measurement, taken from the centroid of the geometry	
h3Index	string	false	A hexadecimal string representing an H3 polygon at resolution 5, derived from the latitude and longitude	
australianMarineRegionsTags	string	true	The australian marine regions tags found applicable to the measurement	
occurrenceStatus	string	false	The presence or absence of the species	

			(`present` or `absent`)	
organismQuantity	float32	true	The biomass of the species of the measurement (if applicable, most studies only measure presence)	gdw/m^2
organismQuantityType	string	true	The type of quantification system used (`biomass`)	
scientificName	string	false	The scientific name of the measurement, as per WoRMS catalogue	
scientificNameID	int64	false	The scientific name aphia id of the measurement, as per WoRMS catalogue	
taxonRank	string	false	The taxon rank of the measurement, as per WoRMS catalogue	
taxonomicStatus	string	false	The taxon acceptance of the measurement, as per WoRMS catalogue	
_temporal_extent	date	false	Derived date used for parquet partitioning (`startYear` - `startMonth??1` - `startDay??1`). **Caution:** for analysis sensitive to the exact day of an occurrence, use `startYear` , `startMonth` , and `startDay` directly.	

3.3 Load

Finally, the transformed Parquet is loaded to the `processing-stored-bucket` at path `datauplift/seagrass/seagrass.parquet`.

4. Flow Advantages

Reproducibility The original Squidle+ relational schema requires deep database expertise to join and flatten. This ETL procedure provides a reliable, replicable path to a flat Darwin Core standard.

Harmonisation the original seagrass data have unique schemes and data encodings which require per dataset effort to harmonise. The pre-harmonisation of this data into an anlysis read product allows scientists to focus on the analysis instead of data wrangling.

Space Efficiency static geo server exports are reduced from 2gb+ to **~230MB** with modern compression and file formats (Parquet), optimising storage and access

Time Efficiency scheduled updates allow users to access up-to-date aggregated data without re-implementing the aggregation steps and validation

Cloud-native workflows reduce I/O overhead