**Data Quality Assessment Framework (DQAF)**

**Technical Documentation**

**Prepared By: BDR Team**

**Date: January 06, 2025**

**Version: 1.0**

[1. Introduction 2](#_Toc1424875359)

[1.1 Purpose of the document 2](#_Toc112748661)

[1.2 Key features of DQAF 2](#_Toc1057197878)

[2. Overview of DQAF assessment functions 3](#_Toc1785094927)

[2.1. Coordinate Precision 3](#_Toc177552655)

[2.2. Coordinate Location in Australia 3](#_Toc2070566516)

[2.3. Coordinate Completeness 4](#_Toc2119327273)

[2.4. Coordinate Unusual 4](#_Toc283641800)

[2.5. Assess Geospatial Accuracy Precision 4](#_Toc695370952)

[2.6. Assess Duplicate Value Combination 4](#_Toc1470420763)

[2.7. Datum Validation 5](#_Toc1976688127)

[2.8. Date Recency 5](#_Toc366165859)

[2.9. Date Format Validation 5](#_Toc640556956)

[2.10. Date Completeness 6](#_Toc1170104300)

[2.11. Scientific Name Completeness 6](#_Toc1234645523)

[2.12. Datum Completeness 6](#_Toc219705836)

[2.13. Datum Validation 6](#_Toc264879883)

[2.14. Coordinate Outlier Detection IQR Method 6](#_Toc2108745333)

[2.15. Coordinate Outlier Detection Isolation Forest Method 7](#_Toc2071437022)

[2.16. Coordinate Outlier Detection Robust Covariance Method 7](#_Toc1390524846)

[2.17. Coordinate Outlier Detection Z-score Method 7](#_Toc158204042)

[2.18. Date Outlier Detection K-means Method 7](#_Toc2083334965)

[2.19. Date Outlier Detection IQR Method 8](#_Toc1130872646)

[2.20. Scientific Name Validation (Not functional at this time) 8](#_Toc770795401)

[3. DQAF scoring mechanism 9](#_Toc748104399)

[3.1 Understanding the Weighting System 9](#_Toc77930477)

[3.2 How overall scores are calculated 9](#_Toc481549550)

[3.3 Calculating the overall score 10](#_Toc421264321)

[4. Use case labelling in DQAF 10](#_Toc1083892833)

[4.1 Understanding the use case structure 11](#_Toc1594703688)

[4.2 How to define and apply use cases 11](#_Toc1248189051)

[4.3 Example use case definition and labelling 12](#_Toc442773721)

[4.4 Example labelling process 12](#_Toc2084034228)

# 1. Introduction

## 1.1 Purpose of the document

The Data Quality Assessment Framework (DQAF) is a tool designed to assess the quality of RDF datasets in the BDR based on a series of structured functions, allowing users to evaluate and then to apply weighted numerical scores or categorical labels indicating the suitability of the observation records for specific purposes.

This document is aimed at end-users and describes three main aspects of the DQAF functionality: performing assessments using predefined functions, scoring observation records and datasets based on customizable criteria, and labelling each occurrence within a dataset for use-case suitability.

This guide is designed to be straightforward and user-friendly, making it easy for end-users to understand and apply the DQAF.

## 1.2 Key features of DQAF

1.  **Data Quality Assessments:** 20 assessment functions are applied to each occurrence in a dataset, to verify different quality aspects such as completeness, consistency, and accuracy. Based on future requirements, we can edit existing functions or add more.

2.  **Scoring System for each occurrence**: Based on a weighted scoring system, observation records are evaluated and given a cumulative score between 0 and 1, where 0 is lowest quality and 1 is highest quality.

3. **Use Case Labels**: Each occurrence is labelled according to its applicability to specific use cases, to help identify which data are most relevant for a particular use case. Labelling for use cases allows users to categorize data based on its relevance and suitability for specific applications. This ensures that high-quality data is used where it is most impactful, and lower-quality data is appropriately flagged for improvement or limited use. Examples of labels used are “Baseline-SDMFFP1” (highest quality for Baseline SDM use case), “Baseline-SDMFFP2” (medium quality) and “Baseline-SDMFFP3” (lowest quality).

This document ensures users can confidently apply the DQAF to meet their needs and tailor it to their specific data quality requirements.

# 2. Overview of DQAF assessment functions

This section provides an overview of the functions used in BDR’s DQAF to evaluate data quality, focusing on attributes such as coordinate precision, completeness, unusual coordinate patterns, location, outlier detection, date validation, scientific name, and datum information.

## 2.1. Coordinate Precision

* **Definition**: Assesses the precision of coordinates by a simple check of how many decimal places are provided. Higher decimal places indicate greater precision.
* **Levels**:
* DQAF.coordinate\_precision:Low: Latitude or longitude has fewer than 2 decimal points, or location data is missing.
* DQAF.coordinate\_precision:Medium: Latitude and longitude have between 2 to 4 decimal points.
* DQAF.coordinate\_precision:High: Latitude and longitude have more than 4 decimal points.

## 2.2. Coordinate Location in Australia

* **Definition**: Verifies whether a given coordinate falls within the geographical boundaries of terrestrial Australia (including off-shore islands), returning the relevant state if applicable. This function uses GEODATA COAST 100K 2004 maps.
* **Levels**:
* DQAF.coordinate\_in\_australia\_state:New\_South\_Wales.
* DQAF.coordinate\_in\_australia\_state:Victoria.
* DQAF.coordinate\_in\_australia\_state:Queensland.
* DQAF.coordinate\_in\_australia\_state:Western\_Australia.
* DQAF.coordinate\_in\_australia\_state:South\_Australia.
* DQAF.coordinate\_in\_australia\_state:Tasmania.
* DQAF.coordinate\_in\_australia\_state:Northern\_Territory.
* DQAF.coordinate\_in\_australia\_state:Australian\_Capital\_Territory.
* DQAF.coordinate\_in\_australia\_state:Outside\_Australia: The occurrence is outside of Australia.

## 2.3. Coordinate Completeness

* **Definition**: Checks if both latitude and longitude are provided for observation records in a dataset.
* **Levels**:
* DQAF.coordinate\_completeness:empty: Coordinate information (latitude/longitude) is missing.
* DQAF.coordinate\_completeness:non\_empty: Coordinate information is provided and non-empty.

## 2.4. Coordinate Unusual

* **Definition**: Identifies any unusual or unexpected coordinate values. The detect\_unusual\_numbers function flags coordinates as unusual if the decimal part contains repeating patterns, such as:
* 45.111111 → Detected as unusual (repeating 1).
* 45.123123123 → Detected as unusual (repeating 123).
* 43. 145145145→ Detected as unusual (repeating 145).
* **Levels**:
* DQAF.coordinate\_unusual:usual: The coordinate does not have an unusual decimal point pattern.
* DQAF.coordinate\_unusual:unusual: The coordinate has an unusual decimal point pattern.

## 2.5. Assess Geospatial Accuracy Precision

* **Definition**: Evaluates the spatial accuracy of the coordinates by comparing them with a predefined level of precision.
* **Levels**:
* DQAF.geo\_spatial\_accuracy\_precision:low\_precision: Indicates that the coordinate uncertainty is either empty or exceeds 10,000 meters, suggesting a low precision for the location.
* DQAF.geo\_spatial\_accuracy\_precision:high\_precision: Assigned when the coordinate uncertainty is 10,000 meters or less, indicating high precision for the location.

## 2.6. Assess Duplicate Value Combination

* **Definition**: Checks if there are duplicate entries based on the combination of selected attributes (e.g., coordinates, date, and other attributes).
* **Levels**:
* DQAF.duplicate\_value\_combination:inferred\_nonduplicate: Label assigned to the first occurrence of a unique combination of values or non-duplicate occurrence s.
* DQAF.duplicate\_value\_combination:inferred\_duplicate: Assigned to duplicate entries found in the dataset based on identical combinations of key attributes.

## 2.7. Datum Validation

* **Definition**: Validates the datum (e.g., WGS84) used in the dataset to ensure consistency across all entries.
* **Levels**:
* DQAF.datum\_type:AGD84: The coordinate datum type is AGD84.
* DQAF.datum\_type:GDA2020: The coordinate datum type is GDA2020.
* DQAF.datum\_type:GDA94: The coordinate datum type is GDA94.
* DQAF.datum\_type:WGS84: The coordinate datum type is WGS84.
* DQAF.datum\_type:None: No datum type is provided.

## 2.8. Date Recency

* **Definition**: Evaluates the recency of dates of the observation records, identifying whether they are up-to-date or outdated compared with a predefined threshold value.
* **Levels**:
* DQAF.date\_recency:recent\_20\_years: The occurrence date falls within the last 20 years, categorized as "recent."
* DQAF.date\_recency:outdated\_20\_years: The occurrence date is older than 20 years, categorized as "outdated."

## 2.9. Date Format Validation

* **Definition**: Checks if dates are formatted correctly according to a specified consistent pattern (e.g., YYYY-MM-DD). The framework also identifies common issues like incorrect formats (e.g., YYYY-DD-MM instead of YYYY-MM-DD) but does not currently validate time or timezone information. For datasets requiring time or timezone validation, additional preprocessing might be necessary.
* **Levels**:
* DQAF.date\_format\_validation:valid: The date is in a valid format.
* DQAF.date\_format\_validation:invalid: The date is not in a valid format.

## 2.10. Date Completeness

* **Definition**: Assesses if all required date fields are populated in the dataset.
* **Levels**:
* DQAF.date\_completeness:empty: Date information is missing in one or more necessary fields.
* DQAF.date\_completeness:non\_empty: Date information is non-empty in all necessary fields.

## 2.11. Scientific Name Completeness

* **Definition**: Assesses if scientific names are provided for all entries in the dataset.
* **Levels**:
* DQAF.scientific\_name\_completeness:empty\_name: The scientific name is missing.
* DQAF.scientific\_name\_completeness:non\_empty\_name: The scientific name is provided.

## 2.12. Datum Completeness

* **Definition**: Assesses if the datum field is filled for all records.
* **Levels**:
* DQAF.datum\_completeness:empty: Datum information is missing.
* DQAF.datum\_completeness:not\_empty: Datum information is provided.

## 2.13. Datum Validation

* **Definition**: Validates the correctness of the provided datum values against expected standards (e.g., WGS84, NAD83).
* **Levels**:
* DQAF.datum\_validation:valid: The datum is valid.
* DQAF.datum\_validation:invalid: The datum is invalid.

## 2.14. Coordinate Outlier Detection IQR Method

* **Definition**: Uses the Interquartile Range (IQR) method to detect outliers in coordinate data, marking them as anomalies.
* **Levels**:
* DQAF.coordinate\_outlier\_irq:outlier\_coordinate: The coordinate is flagged as an outlier using the IQR method.
* DQAF.coordinate\_outlier\_irq:normal\_coordinate: The coordinate is considered normal using the IQR method.

## 2.15. Coordinate Outlier Detection Isolation Forest Method

* **Definition**: Applies an Isolation Forest algorithm to identify outlier coordinates, marking them as anomalies.
* **Levels**:
* DQAF.coordinate\_outlier\_isolation\_forest:outlier\_coordinate: The coordinate is flagged as an outlier using the Isolation Forest algorithm.
* DQAF.coordinate\_outlier\_isolation\_forest:normal\_coordinate: The coordinate is considered normal using the Isolation Forest algorithm.

## 2.16. Coordinate Outlier Detection Robust Covariance Method

* **Definition**: Uses a Robust Covariance technique to find coordinate outliers by measuring how far they deviate from typical values.
* **Levels**:
* DQAF.coordinate\_outlier\_robust\_covariance:outlier\_coordinate: The coordinate is flagged as an outlier using the Robust Covariance method.
* DQAF.coordinate\_outlier\_robust\_covariance:normal\_coordinate: The coordinate is considered normal using the Robust Covariance method.

## 2.17. Coordinate Outlier Detection Z-score Method

* **Definition**: Employs the Z-score method to detect outliers by measuring how far coordinates are from the mean in standard deviations.
* **Levels**:
* DQAF.coordinate\_outlier\_zscore:outlier\_coordinate: The coordinate is flagged as an outlier using Z-score analysis.
* DQAF.coordinate\_outlier\_zscore:normal\_coordinate: The coordinate is considered normal using Z-score analysis.

## 2.18. Date Outlier Detection K-means Method

* **Definition**: Clusters date entries using the K-means algorithm and identifies dates that fall outside typical clusters as outliers.
* **Levels**:
* DQAF.date\_outlier\_kmeans:outlier\_date: The date is flagged as an outlier using the K-means clustering method.
* DQAF.date\_outlier\_kmeans:normal\_date: The date is considered normal using the K-means clustering method.

## 2.19. Date Outlier Detection IQR Method

* **Definition**: Uses the Interquartile Range (IQR) method to detect outliers in date data, identifying anomalous date entries.
* **Levels**:
* DQAF.date\_outlier\_irq:outlier\_date: The date is flagged as an outlier using the IQR method.
* DQAF.date\_outlier\_irq:normal\_date: The date is considered normal using the IQR method.

## 2.20. Scientific Name Validation (Not functional at this time)

* **Definition**: A placeholder function intended for validating scientific names, currently non-functional.
* **Levels**:
* DQAF.scientific\_name\_validation:valid\_name: The scientific name is valid based on reference taxonomy.
* DQAF.scientific\_name\_validation:invalid\_name: The scientific name is invalid.

# 3. DQAF scoring mechanism

The scoring process in DQAF evaluates the quality of each occurrence by applying weighted scores to the results of the assessment functions described above. Each function’s result has an associated weight, allowing for overall scoring to be customised as needed to prioritize different aspects of data quality. The weights used here, found under the BDR\_General\_Weight column in the assertions\_score\_weighting\_definition.xlsx file, serve as an example. Applications or other users of the DQAF can define their unique scoring criteria, adjusting weights to fit specific data quality needs. Based on these custom-defined weights, each occurrence will receive a unique quality score.

## 3.1 Understanding the Weighting System

In the assertions\_score\_weighting\_definition.xlsx file:

- Each specific assessment outcome (e.g., coordinate\_precision:Low, coordinate\_precision:High) has a default assigned weight indicating its impact on the overall score. These default values can be changed in the spreadsheet.

- a higher weight indicates higher importance assigned to a metric.

- users may modify the BDR\_General\_Weight values or define new columns for alternative scoring strategies tailored to specific applications.

For example, if **coordinate\_precision:Low** has a default weight of 0.2.   and **coordinate\_precision:High** has a default weight of 0.5, observation records with high coordinate precision will contribute more to the final score than those with low precision.

## 3.2 How overall scores are calculated

1. **Run assessment functions**: Each data occurrence is evaluated using all assessment functions, resulting in individual results.
2. **Apply weights**: For each result, multiply the result by the corresponding weight.
3. **Aggregate scores**: Sum the weighted values to produce a total quality score for each occurrence.

**Example of scoring calculation**

Consider an occurrence with the following assessment results and weights:

|  |  |  |
| --- | --- | --- |
| **Assessment Function** | **Outcome** | **Weight** |
| coordinate\_precision:High | coordinate\_precision:High | 0.5 |
| coordinate\_completeness:non\_empty | coordinate\_completeness:non\_empty | 0.0 |
| location\_outlier:None | location\_outlier:None | 0.3 |
| date\_validity:Valid | date\_validity:Valid | 0.4 |

## 3.3 Calculating the overall score

Each function’s contribution is determined by multiplying the function’s result (as a numerical value) by its weight.

1. **coordinate\_precision:High** (Weight: 0.5)

* The occurrence meets this outcome, receiving the full weight of 0.5.

1. **coordinate\_completeness:non\_empty** (Weight: 0.0)

* This outcome has a weight of 0.0, so it does not add to the score.

1. **location\_outlier:None** (Weight: 0.3)

* The occurrence meets this outcome, contributing 0.3.

1. **date\_validity:Valid** (Weight: 0.4)

* The occurrence meets this outcome, contributing 0.4.

The total score for this occurrence is calculated as follows: [(0.5) + (0.0) + (0.3) + (0.4) = 1.2]

**Interpreting the scores**

Once calculated, the total score can be interpreted based on quality thresholds, such as:

- **High Quality**: Score ≥ 0.8

- **Medium Quality**: 0.5 ≤ Score < 0.8

- **Low Quality**: Score < 0.5

These thresholds help categorize data observation records according to overall quality, enabling users to filter or prioritize observation records based on specific use case requirements.

# 4. Use case labelling in DQAF

The DQAF framework includes a flexible system for labelling data observation records according to specific use cases. Users or teams can define these use cases, such as Baseline-SDMFFP1, Baseline-SDMFFP2, or custom labels, in the usecase\_definition.xlsx file. This allows applications to filter or categorize biodiversity data based on its suitability for particular requirements, such as research, conservation, or species tracking.

## 4.1 Understanding the use case structure

In the usecase\_definition.xlsx file:

- Each **use case** (e.g., Baseline-SDMFFP1, Baseline-SDMFFP2) is represented by a column.

Each use case category in the usecase\_definition.xlsx file represents a suitability level for specific applications based on data quality requirements. For example:

* Baseline-SDMFFP1: High suitability for baseline species distribution modeling, requiring high-quality data with precise coordinates and valid scientific names.
* Baseline-SDMFFP2: Medium suitability for baseline species distribution modeling, suitable for use cases with moderate data quality.
* Baseline-SDMFFP3: Low suitability for baseline species distribution modeling, acceptable for exploratory analyses or cases with limited data accuracy.

- Each row represents a **data quality assertion** (e.g., coordinate\_precision:Low, coordinate\_completeness:non\_empty).

- Cells contain either a 1 (indicating that the data quality assertion is required for the use case) or are left blank (indicating irrelevance).

For example:

- coordinate\_precision:High has a value of 1 for both Baseline-SDMFFP1 and Baseline-SDMFFP2, indicating that high coordinate precision is required for these use cases.

- Users can add new use cases by creating additional columns and marking relevant assertions with 1s to represent essential criteria.

## 4.2 How to define and apply use cases

1. **Define new use cases**: Add a new column for each use case in usecase\_definition.xlsx and populate it with 1s to represent essential data quality assertions.

* Example: To create a use case category called Conservation\_FFP1, add a column titled Conservation\_FFP1 and mark 1 on the combination of assertions required for this use case, such as coordinate\_precision:High and scientific\_name\_validity:Valid.

1. **Evaluate each occurrence**: During the assessment, each data occurrence is evaluated based on matches with the assertions marked in the selected use case.
2. **Label observation records**: For each occurrence, check if it meets the criteria for the chosen use case (i.e., if it fulfills all marked assertions with 1s in that column). Based on the evaluation, assign a label such as suitable (all marked assertions matched) or not suitable (not all marked assertions matched).

## 4.3 Example use case definition and labelling

Consider the following partial structure in usecase\_definition.xlsx for two use cases in biodiversity studies:

|  |  |  |
| --- | --- | --- |
| **Data quality assertion** | **Baseline-SDMFFP1** | **Baseline-SDMFFP2** |
| coordinate\_precision:High | 1 | 1 |
| coordinate\_completeness:non\_empty | 1 | 1 |
| location\_outlier:None | 1 |  |
| scientific\_name\_validity:Valid | 1 | 1 |
| date\_validity:Valid |  | 1 |

* **Baseline-SDMFFP1** is a general use case requiring high coordinate precision, complete coordinate data, no location outliers, and valid scientific names.
* **Baseline-SDMFFP2** is stricter, additionally requiring valid dates for observation records but does not require the location\_outlier check.

## 4.4 Example labelling process

Assume an occurrence data with the following quality outcomes:

* **coordinate\_precision:High** (meets requirement)
* **coordinate\_completeness:non\_empty** (meets requirement)
* **location\_outlier:None** (meets requirement)
* **scientific\_name\_validity:Valid** (meets requirement)
* **date\_validity:Missing** (does not meet requirement)

For this occurrence:

- It would be labelled as **suitable** for Baseline-SDMFFP1 since all required assertions for this use case are met.

- It would be labelled as **not suitable** for Baseline-SDMFFP2 because it fails the date\_validity:Valid requirement.

This labelling indicates that the occurrence is suitable for conducting basic species distribution modelling, but not for specific applications that need accurate date information, such as tracking seasonal migration patterns.

**Customizing use case requirements**

The flexible structure of DQAF’s use case labelling system allows users to adapt to diverse requirements in biodiversity data analysis. For example, users can:

* **Modify existing use cases**: Update any use case by adjusting the 1s in the corresponding column to include or exclude certain assertions.
* **Add complex use cases**: Define multiple use cases for layered analysis. For example, you might want to separately label observation records for “Flora Conservation” and “Fauna Tracking” to apply different criteria based on plant or animal data.