



OPEN AQUARIUM PROJECT SPECIFICATION

Data Model

PRE-DRAFT

Revision	0.4
Status	Pre-Draft. A candidate document submitted for consideration.
Date	2026-10-18
Identifier	data-model
Authors	Trent Scholl, Project Lead, Open Aquarium Project
Editors	Trent Scholl
License	Community Specification License 1.0

Pre-Draft. This is a proposed draft. The Working Group has not approved it, and it will change. Comment by opening an issue with the `data-model` label.

1 License

This Specification is subject to the Community Specification License 1.0 (the “Specification License”), available at <https://github.com/CommunitySpecification/1.0> and in `LICENSES/Community-Spec-1.0.md`. Contributions to this Specification are made under the Community Specification Contributor License Agreement 1.0.

The JSON Schema, registry, and example files published with this Specification are licensed under the Apache License 2.0 (`LICENSES/Apache-2.0.txt`).

The Scope that bounds the patent commitments made under the Specification License is set out in `Scope.md`. Exclusion notices, license acceptances, and withdrawals are recorded in `Notices.md`.

Notes:

1. Appendices are informative (non-normative) unless an appendix is explicitly marked normative.

THESE MATERIALS ARE PROVIDED “AS IS.” The Contributors and Licensees expressly disclaim any warranties (express, implied, or otherwise), including implied warranties of merchantability, non-infringement, fitness for a particular purpose, or title, related to the materials. The entire risk as to implementing or otherwise using the materials is assumed by the implementer and user. IN NO EVENT WILL THE CONTRIBUTORS OR LICENSEES BE LIABLE TO ANY OTHER PARTY FOR LOST PROFITS OR ANY FORM OF INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER FROM ANY CAUSES OF ACTION OF ANY KIND WITH RESPECT TO THIS DELIVERABLE OR ITS GOVERNING AGREEMENT, WHETHER BASED ON BREACH OF CONTRACT, TORT (INCLUDING NEGLIGENCE), OR OTHERWISE, AND WHETHER OR NOT THE OTHER MEMBER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

2 Compliance with Open Aquarium Tenets

2.1 Openness

The Specification is published under the Community Specification License 1.0, and its schema, registry, and examples under the Apache License 2.0. Anyone may implement it, in open source or proprietary products, without permission or payment. The registry (Section 9) is a public, versioned JSON file that implementers can copy directly into their products. Livestock are identified through public taxonomic databases that anyone can read without an account (Section 8.10).

2.2 Interoperability

Every value carries a parameter identifier and a unit from a shared registry, so a value written by one product means the same thing when read by another. Unit conversions are defined once, in this Specification, instead of being reimplemented by each product. The model is independent of transport and storage, so the same objects serve file export, laboratory results, and the records that controllers keep. Revisions 0.2 to 0.4 add what owners lose today when they change products: sensor history in compact form, the states of equipment, their target ranges, their livestock, their maintenance tasks, whether by date or by hours of running, calibration records, and alarm history. Specifications for kinds of equipment, such as Dosing and Water Testing, record their history with the objects of this Specification rather than formats of their own (Table 11).

2.3 Local Control

The data model is a file format. Reading or writing it does not require a network service, an account, or any vendor. Owners can export their complete history from one product and import it into another, so losing years of records is no longer a reason to stay with a product. Configuration moves in the Automation specification's

documents, which use the same identifiers (Section 8.6), and one export file holds the history and the configuration together, so that moving to a new product is one step (Section 7.8).

2.4 Safety

Unit and basis errors cause real harm in aquariums. Nitrate expressed as nitrogen and nitrate expressed as NO₃ differ by a factor of 4.4, and dKH and meq/L differ by a factor of 2.8. A dosing decision based on a misread value can kill livestock. This Specification requires every value to carry an explicit unit, encodes the chemical basis in the unit code, and excludes ambiguous units such as “ppm” (Section 9.4). Where a test kit does not state its basis, the producer records the unit it assumed and marks the value, and no product acts on the value until a person has confirmed its basis (Section 9.4). A Document records only what has happened, so no time in it is more than a minute later than when it was written: a result with a mistyped date cannot stay the latest result for months (Section 7.4). Specific gravity records the temperature it is referred to, and ORP the reference electrode it is measured against, so that values from different instruments are not compared as if they were the same (Section 8.4). Ranges are the owner’s own statement of what is safe for their livestock: this Specification gives no default ranges, and a product does not record one the owner has not set (Section 8.9). Calibration records keep the evidence of a probe’s condition, so that a failing probe can be found before it misleads control (Section 8.12). An export file is checked before anything in it is used, so that a file from elsewhere cannot write outside the place it is unpacked to or fill a disk (Sections 7.8 and 11). This Specification does not control equipment, so it introduces no electrical or control hazards.

2.5 Sustainability

Portable history lets owners repair, replace, or mix equipment from different makers without losing records, which removes a reason to replace working equipment. Equipment can carry its expected end of life, so that owners replace a worn probe before it fails rather than the controller around it (Section 8.3). Livestock records state whether animals were collected from the wild or bred in captivity (Section 8.10), and RO/DI records state how much water went to the drain (Section 8.6). The format is plain JSON and depends on no proprietary software.

Revision History

Revision	Date	Description	Authors
0.1	2026-10-13	Initial Pre-Draft.	Trent Scholl
0.2	2026-10-15	Series for frequent readings; State Series for the states of equipment; Ranges; Livestock, with taxonomic identifiers; Tasks; Calibration Records; event types for alarms, restarts, refills, RO/DI runs, dosing adjustments, analyzer tests, livestock, modes, overrides, and programs, and events that total several occurrences; guidance for results with no stated chemical basis; the reference temperature of specific gravity and the reference electrode of ORP; corrected and reference observations; the <code>source</code> vessel role, vessel dimensions, and equipment end of life; product GTINs; the parameters <code>free-ammonia</code> , <code>daily-light-integral</code> , and <code>turnover</code> ; vocabulary tables generated from the registry. Documents of revision 0.1 remain valid (Section 7.7).	Trent Scholl
0.3	2026-10-16	<code>registry.json</code> governs where it and this document differ, as the Project Lead decided; reservoirs without <code>circulating</code> are not circulating, while Documents of revisions 0.1 and 0.2 keep the old default, and the validator and library warn when a reservoir leaves it out; Tasks that fall due after a number of hours of running, <code>everyHours</code> ; a total over a day as the specification of the equipment defines it, such as a dosing day that crosses midnight; no time later than the Document’s <code>created</code> , and consumers ask before importing one later than their clock; no product acts on a value with an assumed basis until a person confirms it, which is recorded by writing the value in the confirmed unit; the <code>system</code> , <code>vessel</code> , and <code>equipment</code> of an alarm present in the Document; the equipment category <code>alarm-signal</code> , the units <code>umol/s</code> and <code>%/degC</code> , and the reference electrode <code>ag-agcl-3.5m</code> ; the potentials of reference electrodes from the USGS National Field Manual; the method types <code>titration</code> and <code>photometer</code> widened to automatic analyzers; a Calibration Record’s <code>gain</code> stated to be an observation correction’s <code>factor</code> ; the attribute of State Series that record whether an output runs; a test procedure, mapped to its requirements, for every requirement; references cited by tag; one sentence per line;	Trent Scholl

Revision	Date	Description	Authors
		and a status for every open issue. Documents of revisions 0.1 and 0.2 remain valid (Section 7.7).	
0.4	2026-10-18	The Project Lead's decisions of 17 October 2026 applied: an export file, a ZIP file with a manifest that holds a Document, the Automation configuration document, the alarm settings document, and, for a person who may view it, the audit log (Section 7.8); a converted value rounded to the original's resolution, converted, so that 8.3 dKH is 2.96 meq/L; times up to 60 s later than <code>created</code> accepted, and <code>created</code> REQUIRED; a completed Calibration Record without a <code>result</code> for equipment that reports none, and a Calibration Record's <code>time</code> , for records that no event holds; a configuration document read by the reader's revision, and one written before revision 0.3 upgraded as a Document is; the equipment category <code>fan</code> ; the volume, mass, and length units given a factor to the base unit of their kind; the schema's <code>\$id</code> https://openaquariumproject.org/schemas/data-model/0.4 ; tests that name requirements by section and numbered item, as the convention of every specification. Also: what an upgrading consumer does with a Document that breaks a rule of a later revision; the <code>id</code> of an alarm recorded once for each system an exception to unique identifiers; the hours of running of equipment whose outputs report no <code>operatingHours</code> ; "A pass criterion that checks what a SHOULD asks applies to a product that does it"; corrected test procedures; references to the final revisions of the other specifications; issues 7, 8, 27, 29, 30, and 36 deferred beyond revision 1.0. Documents of revisions 0.1 to 0.3 remain valid (Section 7.7).	Trent Scholl

Contents

1	License	2
2	Compliance with Open Aquarium Tenets	2
2.1	Openness	2
2.2	Interoperability	2
2.3	Local Control	2
2.4	Safety	3
2.5	Sustainability	3
	Revision History	3
3	Scope	7
4	References	7
4.1	Normative References	7
4.2	Informative References	8
5	Terms, Definitions, and Conventions	8
5.1	Requirements Language	8
5.2	Terms and Definitions	8
5.3	Notation	9
6	Introduction	9
6.1	Purpose	9
6.2	Design Principles	10
6.3	Model Overview	10
7	Document Structure	11
7.1	Encoding	11
7.2	Top-Level Object	11
7.3	Identifiers and References	11
7.4	Dates, Timestamps, and Recurrences	12
7.5	Quantity	12
7.6	Extensions	12
7.7	Revisions and Compatibility	13
7.8	Export Files	14
8	Objects	15
8.1	System	15
8.2	Vessel	15
8.3	Equipment	16
8.4	Observation	16
8.5	Sample	18
8.6	Event	18
8.7	Series	20
8.8	State Series	21
8.9	Range	22
8.10	Livestock	23
8.11	Task	24
8.12	Calibration Record	25
8.13	Alarm	27
9	Registry	27
9.1	General	27
9.2	Parameters	27
9.3	Elements	29
9.4	Units	29
9.5	Unit Conversion	31
9.6	Extension Parameters and Units	32
9.7	Vocabularies	32
10	Conformance	36
10.1	Producers	36
10.2	Consumers	36
10.3	Schema and Validation Tools	36
10.4	Test Procedures	37
11	Security and Privacy Considerations	39

Appendix A — Approval Checklist	40
Appendix B — Examples (Informative)	40
Appendix C — Open Issues (Informative)	41

PRE-DRAFT

3 Scope

This Specification defines a data model and a JSON encoding for describing:

- aquatic systems kept by people, such as aquariums and ponds, and the vessels that make them up;
- equipment associated with those systems, and its calibrations;
- measurements of water and its environment, one at a time or as series;
- the states of equipment over time, such as when an outlet was on or a float switch was wet;
- water samples analyzed by test kits, instruments, or laboratories;
- events such as water changes, doses, alarms, and maintenance;
- the livestock kept in systems;
- owners' target ranges for parameters;
- maintenance tasks and when they fall due; and
- export files, which hold a Document together with the configuration documents of other specifications.

This Specification does not define:

- how devices communicate or are controlled, which the Device Protocol, Device Classes, and Automation specifications define;
- controller configuration, which the Automation specification's configuration documents carry [OAP-AU Section 7];
- how products store data internally;
- recommended values or ranges for any parameter; or
- how products remind people or notify them of alarms, which the Alarms and Notifications specification defines.

This revision is intended for moving history and configuration between products, publishing laboratory results in a portable form, recording manual test results, and keeping the records of controllers: sensor history, the states of equipment, alarms, calibrations, and maintenance.

4 References

4.1 Normative References

- **[RFC 2119]** Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- **[RFC 8174]** Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017.
- **[RFC 8259]** Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, December 2017.
- **[RFC 3339]** Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.
- **[RFC 6838]** Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, January 2013.
- **[JSON Schema]** "JSON Schema: A Media Type for Describing JSON Documents" and "JSON Schema Validation", draft 2020-12, <https://json-schema.org/draft/2020-12>.
- **[ZIP]** ISO/IEC 21320-1:2015, "Information technology — Document Container File — Part 1: Core", <https://www.iso.org/standard/60101.html>. It defines the ZIP files that export files are (Section 7.8), and is available free of charge.
- **[OAP-AN]** Open Aquarium Project, "Open Aquarium Alarms and Notifications", Revision 0.3, [specs/alarms-and-notifications/spec.md](https://open-aquarium-project.github.io/specs/alarms-and-notifications/spec.md). Its Table 25 defines the alarms that events hold (Section 8.13), and its alarm settings documents go in export files (Section 7.8).
- **[OAP-AU]** Open Aquarium Project, "Open Aquarium Automation", Revision 0.3, [specs/automation/spec.md](https://open-aquarium-project.github.io/specs/automation/spec.md). Its configuration documents go in export files (Section 7.8).
- **[OAP-DC]** Open Aquarium Project, "Open Aquarium Device Classes", Revision 0.5, [specs/device-classes/spec.md](https://open-aquarium-project.github.io/specs/device-classes/spec.md). Its attributes name what State Series record (Section 8.8), and its `operatingHours` counts the running of Tasks' equipment (Section 8.11).

- **[OAP-SA]** Open Aquarium Project, “Open Aquarium Security and Access”, Revision 0.3, [specs/security-and-access/spec.md](#). Its audit log, and the operations that may export it, go in export files (Section 7.8).

4.2 Informative References

- **[RFC 9562]** Davis, K., Peabody, B., and P. Leach, “Universally Unique IDentifiers (UUIDs)”, RFC 9562, May 2024.
- **[IUPAC]** IUPAC Periodic Table of the Elements and standard atomic weights, <https://iupac.org/what-we-do/periodic-table-of-elements/>.
- **[UNESCO-44]** UNESCO, “Algorithms for computation of fundamental properties of seawater”, Fofonoff, N. P. and Millard, R. C., Unesco Technical Papers in Marine Science 44, 1983, <https://unesdoc.unesco.org/ark:/48223/pf0000059832>. Its equation of state of seawater gives the densities from which Section 8.4 computes specific gravity.
- **[USGS]** Nordstrom, D. K. and Wilde, F. D., “Reduction-Oxidation Potential (Electrode Method)”, Section 6.5 of Chapter A6 of the U.S. Geological Survey’s “National Field Manual for the Collection of Water-Quality Data”, Techniques of Water-Resources Investigations, Book 9, Version 1.2, September 2005, <https://doi.org/10.3133/twri09A6.5>. Its Table 6.5–2 gives the potentials of silver chloride reference electrodes against the standard hydrogen electrode, which Table 27 rounds.
- **[GS1]** GS1, “GS1 General Specifications”, <https://www.gs1.org/standards/barcodes-epcrfid-id-keys/gs1-general-specifications>. It defines the Global Trade Item Number and its check digit.
- **[WoRMS]** WoRMS Editorial Board, “World Register of Marine Species”, <https://www.marinespecies.org/>.
- **[GBIF]** GBIF Secretariat, “GBIF Backbone Taxonomy”, <https://www.gbif.org/dataset/d7dddbf4-2cf0-4f39-9b2a-bb099caae36c>.
- **[ITIS]** “Integrated Taxonomic Information System”, <https://www.itis.gov/>.
- **[COL]** “Catalogue of Life”, <https://www.catalogueoflife.org/>.
- **[OAP-DO]** Open Aquarium Project, “Open Aquarium Dosing”, Revision 0.3, [specs/dosing/spec.md](#).
- **[OAP-FP]** Open Aquarium Project, “Open Aquarium Flow Programs”, Revision 0.3, [specs/flow-programs/spec.md](#).
- **[OAP-FS]** Open Aquarium Project, “Open Aquarium Fail-Safe Behavior”, Revision 0.4, [specs/fail-safe-behavior/spec.md](#). Its bound on the error of a valid device clock gives the 60 s by which a time may be later than `created` (Section 7.4).
- **[OAP-LP]** Open Aquarium Project, “Open Aquarium Lighting Programs”, Revision 0.3, [specs/lighting-programs/spec.md](#).
- **[OAP-PI]** Open Aquarium Project, “Open Aquarium Probe Interface”, Revision 0.4, [specs/probe-interface/spec.md](#).
- **[OAP-WL]** Open Aquarium Project, “Open Aquarium Water Level and Water Change”, Revision 0.3, [specs/water-level-and-water-change/spec.md](#).
- **[OAP-WT]** Open Aquarium Project, “Open Aquarium Water Testing”, Revision 0.3, [specs/water-testing/spec.md](#).

5 Terms, Definitions, and Conventions

5.1 Requirements Language

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “NOT RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in BCP 14 [RFC 2119] [RFC 8174] when, and only when, they appear in all capitals, as shown here.

5.2 Terms and Definitions

System (aquatic system): A body of water kept for aquatic organisms, together with every vessel that shares its water. A display tank with a sump is one system; a quarantine tank with its own filtration is a separate system.

Vessel: A container of water that belongs to a system, such as a display tank, sump, refugium, or pond, or a reservoir or source that supplies water to the system.

Circulating vessel: A vessel that continuously exchanges water with the system's other circulating vessels.

Equipment: A physical device associated with a system, such as a heater, pump, light, doser, sensor, or controller.

Channel: A part of one piece of equipment that can be controlled or measured on its own, such as one head of a multi-head doser.

Parameter: A measurable property of water or its environment, such as temperature, alkalinity, or nitrate.

Observation: One measured or calculated value of one parameter, for one system, at one time.

Series: Many values of one parameter, measured in one place by one method, recorded compactly (Section 8.7).

State Series: The states of one attribute of one piece of equipment over time, recorded as its changes (Section 8.8).

Sample: Water taken from a system at a specific time for analysis, such as water sent to a laboratory.

Event: A recorded action or occurrence affecting a system, such as a water change, a dose, or an alarm.

Range: An owner's desired and acceptable values of one parameter, and the limits beyond which its values are dangerous (Section 8.9).

Livestock: The fish, corals, invertebrates, plants, and other organisms kept in a system.

Taxon: A group of organisms that taxonomists name, such as a species or a genus.

Task: A piece of maintenance that people do by hand, once or on a schedule (Section 8.11).

Calibration: Comparing what a piece of equipment measures or delivers with known references, and adjusting it to agree with them.

Registry: The parameters, units, conversions, and vocabularies defined in Section 9 and published in machine-readable form in `registry.json`.

Extension name: An identifier in reverse domain name notation, such as `com.example.reactor-flow`, used to add information that this Specification does not define.

Document: A JSON text that conforms to this Specification.

Producer: Software or a device that writes Documents.

Consumer: Software or a device that reads Documents.

Configuration document: A document of another specification that carries a controller's configuration, and Data Model objects such as systems and equipment without an `oapDataModel` of its own: an Automation configuration document [OAP-AU Section 7], or an Alarms and Notifications alarm settings document [OAP-AN Section 19].

Export file: A ZIP file that holds a Document together with the configuration documents and audit log that go with it, and a manifest that lists them (Section 7.8).

5.3 Notation

JSON member names appear in `monospace`. The types string, number, boolean, object, and array are the JSON types defined in [RFC 8259]. Types defined by this Specification, such as Identifier and Quantity, are capitalized. In member tables, the Presence column states whether a member is REQUIRED or OPTIONAL.

6 Introduction

6.1 Purpose

Aquarium owners collect data from many sources: sensors attached to a controller, test kits read by eye, handheld photometers, and laboratory analyses. Each product records this data in its own format, with its own

names and units. Owners cannot easily move history between products, and the same number can mean different things in different products. A nitrate result of “10” might be milligrams per litre of nitrate in one product and milligrams per litre of nitrogen in another.

This Specification defines a common model so that:

- an owner can export their full history from one product and import it into another;
- a laboratory can publish results that any product can import; and
- the meaning of every value is explicit: which parameter, which unit, which chemical basis, and which method.

6.2 Design Principles

1. **Explicit units.** Every value carries its unit. There is no default unit.
2. **Basis in the unit.** Where a concentration can be expressed relative to different chemical species, such as nitrate as NO₃ or as N, the unit code states which.
3. **Preserve what was measured.** Producers record values in the unit the instrument reported. Conversion is done by consumers, using the factors in this Specification.
4. **Not reef-only.** Marine, brackish, and freshwater systems are modeled the same way. Nothing in the model assumes a reef.
5. **Open-world extension.** Producers can add parameters, vocabulary values, and data under their own extension names without coordination. Consumers ignore what they do not understand.
6. **Transport-independent.** The model defines content, not transport. The same objects can be stored in files, published by a laboratory, or carried by a device protocol.
7. **One record, many specifications.** Specifications for kinds of equipment record their history with these objects, and keep only what is theirs alone in extensions (Section 8.6).

6.3 Model Overview

A Document contains one or more systems. Each system contains its vessels. Equipment, livestock, observations, series, samples, events, ranges, and tasks refer to a system, and optionally to one of its vessels, by identifier. Observations from a laboratory or test session refer to the sample they were measured from. State Series refer to the equipment whose states they record.

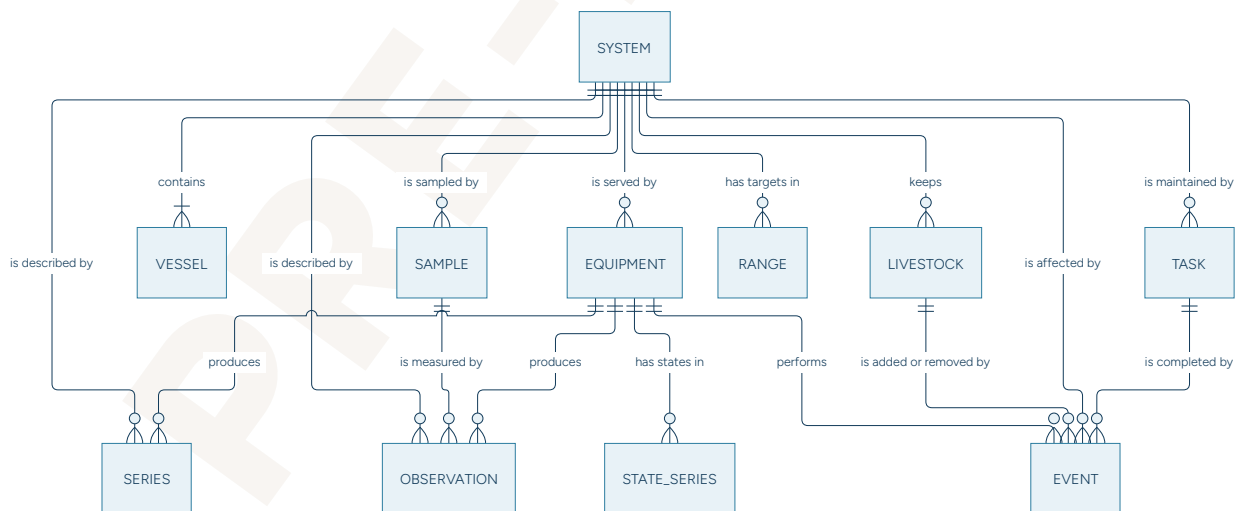


Figure 1 — Objects in a Document and their references

7 Document Structure

7.1 Encoding

A Document **MUST** be a JSON text [RFC 8259] encoded in UTF-8, and its top-level value **MUST** be an object. Member names are case-sensitive. Numeric values **MUST** be finite; consumers **SHOULD** read them as IEEE 754 double-precision numbers.

7.2 Top-Level Object

Table 1 — Top-level members

Member	Type	Presence	Description
<code>oapDataModel</code>	string	REQUIRED	Revision of this Specification that the Document conforms to. For this revision, "0.4". Section 7.7 says how Documents of revisions "0.1" to "0.3" are read.
<code>generator</code>	Generator	OPTIONAL	The product that wrote the Document.
<code>created</code>	Timestamp	REQUIRED	When the Document was written. OPTIONAL in Documents of revisions 0.1 to 0.3. <i>REQUIRED from 0.4.</i>
<code>systems</code>	array of System	REQUIRED	The systems described. MUST contain at least one System.
<code>equipment</code>	array of Equipment	OPTIONAL	Equipment associated with the systems.
<code>livestock</code>	array of Livestock	OPTIONAL	Livestock kept in the systems (Section 8.10).
<code>parameters</code>	array of Parameter Definition	OPTIONAL	Definitions of extension parameters used in the Document (Section 9.6).
<code>observations</code>	array of Observation	OPTIONAL	Measured or calculated values.
<code>series</code>	array of Series	OPTIONAL	Measured values in compact form (Section 8.7).
<code>samples</code>	array of Sample	OPTIONAL	Water samples that observations were measured from.
<code>events</code>	array of Event	OPTIONAL	Events.
<code>states</code>	array of State Series	OPTIONAL	The states of equipment over time (Section 8.8).
<code>ranges</code>	array of Range	OPTIONAL	Owners' target ranges (Section 8.9).
<code>tasks</code>	array of Task	OPTIONAL	Maintenance tasks (Section 8.11).
<code>extensions</code>	Extensions	OPTIONAL	See Section 7.6.

Table 2 — Generator members

Member	Type	Presence	Description
<code>name</code>	string	REQUIRED	Product name.
<code>version</code>	string	OPTIONAL	Product version.
<code>manufacturer</code>	string	OPTIONAL	Product maker.
<code>extensions</code>	Extensions	OPTIONAL	See Section 7.6.

7.3 Identifiers and References

An Identifier is a string of 1 to 128 characters drawn from A–Z, a–z, 0–9, ., _, :, and -.

- Every Identifier used as an `id` member **MUST** be unique within the Document, across all object types. The `id` of a Channel is an exception: it **MUST** be unique only within its Equipment. The `id` of an Alarm is another: an alarm about a controller as a whole is recorded once for each system the controller serves, each time with the same `id` (Section 8.13, item 3).
- A reference member, such as `system` or `vessel`, contains the `id` of another object. That object **MUST** be present in the same Document.
- Producers **SHOULD** assign identifiers that stay the same across exports from the same product, so that consumers can recognize objects they have already imported. UUIDs [RFC 9562] are **RECOMMENDED** for this purpose.

4. The `element` member of an Event is an exception to item 2. It contains the `id` of an element of an Automation configuration document [OAP-AU Section 7.3], which is not part of the Document.
5. The `system`, `vessel`, and `equipment` of an Event's `alarm` (Section 8.13) are references, and item 2 applies to them. The other identifiers of an alarm, those of its `subject` and its `output`, name elements of the controller's configuration and its devices, as `element` does, and need not be present in the Document.

7.4 Dates, Timestamps, and Recurrences

A Timestamp is an RFC 3339 `date-time` [RFC 3339] that includes a UTC offset, such as `2026-10-13T08:00:00-07:00` or `2026-10-13T15:00:00Z`. Producers MUST NOT write a time without an offset. Producers SHOULD use the offset in effect where the system is located, so that consumers can recover the local time of day.

A Document records what has happened.

1. A producer MUST NOT write a time of an observation, a sample's `collected` or `analyzed`, an event's `time` or `end`, a Calibration Record's `time`, the time of a value of a Series or State Series, or a State Series' `end` that is more than 60 s later than the Document's `created`. The 60 s allow for the clocks of the instruments that took the values, which may run ahead of the producer's clock by as much as a valid device clock may err [OAP-FS Section 10.5.4].
2. A consumer MUST NOT act on a value whose time is later than its own clock, and SHOULD ask a person before it imports one. Such a time is usually a wrong date, and a value dated in the future would stay the latest value of its parameter until that date passed.
3. Times that are not of what has happened, such as a Calibration Record's `due` or a Task's `due`, may be later than `created`.

A Date is an RFC 3339 `full-date`, such as `2026-10-13`. Dates are used where only the day matters, such as when a system was established. The local date of a Timestamp is its date in its own UTC offset.

A Recurrence is a string made of `P`, a whole number from 1 to 9999, and a letter: `D` for days, `W` for weeks, `M` for calendar months, or `Y` for calendar years. For example, `P7D` is every seven days, `P2W` every two weeks, and `P3M` every three months. It is a subset of the duration syntax collected in [RFC 3339] Appendix A. Adding months or years to a date that does not exist in the resulting month, such as one month to 31 January, gives the last day of that month.

Durations, and the times within a Series or State Series, are numbers of seconds.

7.5 Quantity

A Quantity is an object with a `value` (number, REQUIRED) and a `unit` (unit code, REQUIRED; Section 9.4). Quantities are used for vessel volumes and dimensions, event amounts, and calibrations. Observations carry `value` and `unit` directly instead of nesting a Quantity, to keep large observation sets compact.

1. Each volume, mass, and length unit of Table 20 has a factor, in `registry.json`, that converts a value in the unit to the base unit of its kind: `L` for volume, `g` for mass, and `m` for length. For example, the factor of `gal` is 3.785411784, and that of `in` is 0.0254.
2. A consumer that converts a Quantity to another unit of the same kind MUST multiply its `value` by the factor of its unit, and divide it by the factor of the other unit. So a consumer can add the volumes of a system's vessels when one is given in `gal` and another in `L` (Section 8.2): 75 gal and 40 L are 323.9 L.

7.6 Extensions

1. Any object defined by this Specification MAY contain an `extensions` member whose value is an object. Each member name in `extensions` MUST be an extension name, and its value MAY be any JSON value.
2. An extension name consists of two or more labels separated by dots. Each label consists of lowercase letters, digits, and hyphens, and does not begin or end with a hyphen. An extension name SHOULD be the reverse of an Internet domain name controlled by the party defining the extension, optionally followed by further labels (for example, `com.example` or `com.example.reactor-flow`).
3. Producers MUST NOT add members that this Specification does not define, except inside `extensions`.

4. Consumers MUST ignore extensions they do not understand. Consumers that write Documents they have read SHOULD preserve extensions they do not understand.
5. Values in the open vocabularies of Section 9.7, extension parameters (Section 9.6), and extension units MAY be extension names.
6. Extension names that begin with `org.openaquariumproject` belong to the specifications of the Open Aquarium Project, which define their meaning, such as `org.openaquariumproject.dosing` [OAP-DO Section 16]. Producers MUST NOT use extension names that begin with `org.openaquariumproject` with any other meaning.

7.7 Revisions and Compatibility

Before revision 1.0, any revision of this Specification may make incompatible changes.

From revision 1.0 onward, a revision with the same major number as an earlier revision will be backward compatible with it. A later minor revision will only add OPTIONAL members, registry parameters, units, and vocabulary values, and a consumer will read a Document that declares a later minor revision of a major revision it implements, ignoring the members, parameters, units, and vocabulary values it does not recognize.

Revisions 0.2 to 0.4 add to the revisions before them: objects, members, parameters, units, and vocabulary values. Revision 0.2 changed the registry file, not Documents: its vocabularies became lists of entries with meanings, rather than lists of names (Section 9.1). Revision 0.3 changes one default: a `reservoir` without `circulating` is not circulating (Section 8.2), where in revisions 0.1 and 0.2 it is. It also adds two rules that Documents of earlier revisions may break: no time later than `created` (Section 7.4), and the objects an alarm names present in the Document (Section 7.3, item 5). Revision 0.4 makes `created` REQUIRED (Section 7.2), and relaxes two rules: it accepts times up to 60 s later than `created` (Section 7.4), and a completed Calibration Record without a `result` from equipment that reports none (Section 8.12).

1. A Document of revision 0.1 is a Document of revision 0.2 once its `oapDataModel` is changed to "0.2". A Document of revision 0.2 is a Document of revision 0.3 once its `oapDataModel` is changed to "0.3", and each of its reservoirs without `circulating` is given `circulating true`, if it also meets the rules that revision 0.3 added. A Document of revision 0.3 is a Document of revision 0.4 once its `oapDataModel` is changed to "0.4", if it has `created`.
2. A consumer that implements revision 0.4 MUST read Documents of revisions 0.1 to 0.3 as item 1 says. It MUST NOT reject such a Document only because it breaks a rule that a later revision added.
3. A consumer that writes what it has read from a Document of an earlier revision as a Document of a later revision MUST make it meet the rules of the later revision. It gives the Document the time it writes it as `created`, asks a person to correct any time later than that (Section 7.4, item 2), and leaves out of an alarm a `system`, `vessel`, or `equipment` that names no object of the Document (Section 7.3, item 5).
4. A Document that declares a revision MUST NOT use what a later revision added. The registry marks its additions with `since` (Section 9.1), and the member tables of Sections 7 and 8 mark the members that revisions 0.3 and 0.4 added or made REQUIRED.
5. Before revision 0.2, other specifications of the project recorded some history under provisional extension names. Table 3 lists those that revision 0.2 replaces. A consumer SHOULD read each as its replacement, and a producer that writes revision 0.2 or later SHOULD write the replacement.
6. Objects of this Specification that configuration documents carry, such as the systems and equipment of an Automation configuration document [OAP-AU Section 7.4], have no `oapDataModel` of their own. A consumer MUST read them as objects of the revision of this Specification that it implements, except those of a configuration document written before revision 0.3, which it MUST first upgrade as item 1 upgrades a Document, so that each of their reservoirs without `circulating` is circulating. The configuration documents written before revision 0.3 are the Automation configuration documents and the alarm settings documents of revision 0.1 of their specifications. A producer SHOULD give every reservoir among them `circulating`, so that they mean the same under every revision.

Table 3 — Provisional extension names that revision 0.2 replaces

Before revision 0.2	Replacement	Where it was defined
Event type <code>org.openaquariumproject.alarm</code> , with the alarm in extensions[" <code>org.openaquariumproject.alarm</code> "]	Event type <code>alarm</code> , with the alarm in <code>alarm</code>	[OAP-AN Section 20.1]
Event type <code>org.openaquariumproject.restart</code> , with the notice in extensions[" <code>org.openaquariumproject.alarm</code> "]	Event type <code>restart</code> , with the notice in <code>alarm</code>	[OAP-AN Section 20.2]
extensions[" <code>org.openaquariumproject.alarm</code> "] of a power-outage event	<code>alarm</code>	[OAP-AN Section 20.2]
Event type <code>org.openaquariumproject.dose-adjusted</code>	Event type <code>dose-adjusted</code>	[OAP-DO Section 16.2]
Event type <code>org.openaquariumproject.container-refill</code>	Event type <code>refill</code>	[OAP-DO Section 16.3]
Event type <code>org.openaquariumproject.analyzer-test</code>	Event type <code>analyzer-test</code>	[OAP-WT Section 15.2]
raw, correction, and reference in extensions[" <code>org.openaquariumproject.water-testing</code> "] of an observation	The observation's correction and reference	[OAP-WT Section 15.1]
Parameter <code>org.openaquariumproject.daily-light-integral</code> , in the unit <code>org.openaquariumproject.mol-per-m2-day</code>	Parameter <code>daily-light-integral</code> , in <code>mol/m2/d</code>	[OAP-LP Section 19]
Parameter <code>org.openaquariumproject.turnover</code> , in the unit <code>org.openaquariumproject.per-hour</code>	Parameter <code>turnover</code> , in <code>1/h</code>	[OAP-FP Section 14.3]

The Dosing, Water Testing, and Probe Interface specifications record calibrations as Calibration Records (Section 8.12), from their revisions 0.2 (Table 11).

7.8 Export Files

An export file moves an owner's history and configuration to another product in one file. It is a ZIP file that holds a Document, the controller's configuration documents, and, for a person who may view it, the audit log of the Security and Access specification, with a manifest that lists them.

1. An export file MUST be a document container file as ISO/IEC 21320-1 defines it [ZIP]: a ZIP file in one part, whose members are stored or compressed with Deflate, none of them encrypted, with names in UTF-8. Its name SHOULD end in `.zip`, so that common tools open it.
2. The name of each member MUST be a relative path: names separated by `/`, none of them empty, `.`, or `...`. Two members MUST NOT have the same name.
3. The member `manifest.json`, at the root of the file, is the manifest. It MUST be a JSON object [RFC 8259] valid against `$defs/exportManifest` of the schema (Section 10.3), with these members:
 - `oapExport` (string, REQUIRED): the revision of this Specification whose export file the manifest follows; for this revision, "`0.4`";
 - `generator` (Generator, OPTIONAL): the product that wrote the file;
 - `created` (Timestamp, REQUIRED): when the file was written;
 - `members` (array, REQUIRED): one entry for each other member of the file, with the member's name as `path` (string, REQUIRED), what it holds as `kind` (string, REQUIRED; item 4), its media type as `mediaType` (string, REQUIRED) [RFC 6838], the revision that it declares as `revision` (string, REQUIRED), and `extensions` (OPTIONAL); and
 - `notes` (string, OPTIONAL) and `extensions` (OPTIONAL). The manifest MUST list every other member of the file, once.
4. A member's `kind` is one of these, or an extension name, for a member that another party defines:
 - `data-model`: a Document, whose revision is its `oapDataModel`;
 - `automation`: an Automation configuration document [OAP-AU Section 7], whose revision is its `oapAutomation`;
 - `alarm-settings`: an alarm settings document [OAP-AN Section 19], whose revision is its `oapAlarms`; and
 - `audit-log`: the audit log, as the Security and Access specification exports it [OAP-SA Section 12.3], whose revision is its `oapAuditLog`.

5. An export file MUST hold one member of the kind `data-model`, and at most one of each other kind of item 4. The `mediaType` of each such member MUST be `application/json` [RFC 8259], and its `revision` MUST be the revision that the document declares.
6. A producer that keeps a configuration, such as a controller, MUST include its Automation configuration document and its alarm settings document. They SHOULD use the same identifiers for systems and equipment as the Document (Section 8.6, item 4).
7. A producer MUST include the audit log only for a person who may view it and export it, with the operations `view-audit-log` and `export-data` [OAP-SA Section 12.3]. The audit log keeps its own format and rules (Section 11).
8. A consumer MUST refuse an export file that breaks items 1 to 5, rather than import part of it. It reads each member of a kind that it implements as the member's specification says, a Document of an earlier revision as Section 7.7 says, and ignores members of other kinds.

For example, the manifest of an export file from a controller:

```
{ "oapExport": "0.4", "created": "2026-10-20T18:30:00+10:00",
  "members": [
    { "path": "history.json", "kind": "data-model", "mediaType": "application/json", "revision": "0.4" },
    { "path": "automation.json", "kind": "automation", "mediaType": "application/json", "revision": "0.3" },
    { "path": "alarm-settings.json", "kind": "alarm-settings", "mediaType": "application/json", "revision": "0.3" } ] }
```

The `export` directory holds an export file, unpacked (Appendix B).

8 Objects

8.1 System

Table 4 — System members

Member	Type	Presence	Description
<code>id</code>	Identifier	REQUIRED	
<code>name</code>	string	REQUIRED	Name shown to the owner.
<code>water</code>	string	REQUIRED	One of <code>marine</code> , <code>brackish</code> , or <code>freshwater</code> . This vocabulary is closed.
<code>tags</code>	array of string	OPTIONAL	Descriptive tags, such as <code>reef</code> or <code>planted</code> . Table 22 lists common tags. Each tag is a registry name or an extension name.
<code>established</code>	Date	OPTIONAL	When the system was first filled with water.
<code>vessels</code>	array of Vessel	REQUIRED	MUST contain at least one Vessel.
<code>notes</code>	string	OPTIONAL	Free text.
<code>extensions</code>	Extensions	OPTIONAL	

8.2 Vessel

Table 5 — Vessel members

Member	Type	Presence	Description
<code>id</code>	Identifier	REQUIRED	
<code>name</code>	string	OPTIONAL	Name shown to the owner.
<code>role</code>	string	REQUIRED	Vessel role (Table 23).
<code>circulating</code>	boolean	OPTIONAL	Whether the vessel is a circulating vessel. Default <code>false</code> for the roles <code>source</code> and <code>reservoir</code> , and <code>true</code> otherwise (see below).
<code>volume</code>	Quantity	OPTIONAL	Estimated volume of water in the vessel, not its nominal size. The unit MUST be a volume unit.
<code>dimensions</code>	object	OPTIONAL	The vessel's inside size and the depth of its water (see below).
<code>notes</code>	string	OPTIONAL	Free text.
<code>extensions</code>	Extensions	OPTIONAL	

The water volume of a system is the sum of the volumes of its circulating vessels. A reservoir, such as a top-off or new-water reservoir, usually does not exchange water with the system, and a system's volume that counted it would be too large.

1. A reservoir is not circulating unless its `circulating` is `true`, as it is for a reservoir that is plumbed into the sump and exchanges water with it all the time.
2. In Documents of revisions 0.1 and 0.2, and in configuration documents written before revision 0.3, a reservoir without `circulating` is circulating (Section 7.7, items 1 and 6).
3. A producer SHOULD write `circulating` for every reservoir, so that it means the same in every revision.

A vessel with the role `source` describes a supply of water that the system does not store, such as tap water, well water, or the product water of an RO/DI unit on its way to a reservoir. It lets producers record measurements of that water, such as the total dissolved solids of RO/DI product water or the nitrate of tap water, as observations of the vessel. A `source` vessel MUST NOT set `circulating` to `true`, and SHOULD NOT have a `volume`. Water that is stored, such as new saltwater mixed for a water change, is a `reservoir` [OAP-WL Section 9.15].

`dimensions` has the members `length`, `width`, and `height`, the inside measurements of the vessel, and `depth`, the depth of its water. Each is a Quantity whose unit MUST be a length unit, and at least one MUST be present. For a vessel that is not rectangular, `length` and `width` are those of the smallest rectangle that encloses its water surface. The Flow Programs specification computes a tank's resonant period from its water's length, width, and depth [OAP-FP Section 8.6].

8.3 Equipment

Table 6 — Equipment members

Member	Type	Presence	Description
<code>id</code>	Identifier	REQUIRED	
<code>system</code>	Identifier	REQUIRED	The system the equipment serves.
<code>vessel</code>	Identifier	OPTIONAL	The vessel where the equipment is located. MUST belong to <code>system</code> .
<code>category</code>	string	REQUIRED	Equipment category (Table 24).
<code>name</code>	string	OPTIONAL	Name shown to the owner.
<code>manufacturer</code>	string	OPTIONAL	
<code>model</code>	string	OPTIONAL	
<code>serialNumber</code>	string	OPTIONAL	See Section 11 before including.
<code>installed</code>	Date	OPTIONAL	When the equipment was put into service.
<code>removed</code>	Date	OPTIONAL	When the equipment was taken out of service.
<code>endOfLife</code>	Date	OPTIONAL	When the equipment is expected to need replacing, such as a pH electrode's predicted end of life [OAP-PI Section 14.10] or the end of a UV lamp's rated life.
<code>channels</code>	array of Channel	OPTIONAL	Parts of the equipment that are controlled or measured separately.
<code>notes</code>	string	OPTIONAL	Free text.
<code>extensions</code>	Extensions	OPTIONAL	

A Channel has an `id` (Identifier, REQUIRED), a `name` (string, OPTIONAL), and `extensions` (OPTIONAL). Producers SHOULD keep equipment that has been removed in exported Documents, with `removed` set, so that earlier observations and events can still refer to it. A producer that predicts `endOfLife`, for example from the trend of a probe's calibrations, SHOULD update it after each calibration.

8.4 Observation

Table 7 — Observation members

Member	Type	Presence	Description
<code>id</code>	Identifier	OPTIONAL	

Member	Type	Presence	Description
parameter	string	REQUIRED	Parameter identifier (Section 9.2, 9.3, or 9.6).
value	number	REQUIRED	The value, in <code>unit</code> .
unit	string	REQUIRED	Unit code. MUST be accepted for <code>parameter</code> .
unitAssumed	boolean	OPTIONAL	<code>true</code> when the method did not state the unit's chemical basis, and the producer assumed it (Section 9.4). Default <code>false</code> .
qualifier	string	OPTIONAL	< or >. See below.
time	Timestamp	see below	When the value applies.
system	Identifier	REQUIRED	The system the value describes.
vessel	Identifier	OPTIONAL	The vessel the value describes. MUST belong to <code>system</code> . When absent, the value describes the system's circulating water.
sample	Identifier	OPTIONAL	The Sample the value was measured from.
method	Method	OPTIONAL	How the value was obtained.
correction	object	OPTIONAL	The correction applied to the value (see below).
reference	boolean	OPTIONAL	<code>true</code> when the value is used as a reference to check or correct other methods, such as a laboratory result used to correct an analyzer [OAP-WT Section 11.2]. Default <code>false</code> .
notes	string	OPTIONAL	Free text.
extensions	Extensions	OPTIONAL	

Requirements:

1. An Observation MUST contain `time`, `sample`, or both. When `sample` is present, the observation applies at the sample's `collected` time. If `time` is also present, it MUST equal the sample's `collected` time.
2. When `sample` is present, the Sample's `system` MUST equal the observation's `system`. If both the observation and the Sample have a `vessel`, they MUST be equal.
3. A `qualifier` of < means the true value is below `value`; `value` is usually the method's detection limit. A `qualifier` of > means the true value is above `value`, for example when a result exceeds the range of a test. Consumers MUST NOT treat a qualified value as an exact value, for example when averaging or charting, and SHOULD display the qualifier with the value.
4. `correction` records that `value` was corrected from what the method reported, for example by a correction derived from laboratory results [OAP-WT Section 11]. It has `raw` (number, REQUIRED), the value as the method reported it, in `unit`; `factor` (number, OPTIONAL, default 1); and `offset` (number, OPTIONAL, default 0), in `unit`. `value` is $raw \times factor + offset$, rounded as the producer rounds its values. `factor` is the multiplier that a Calibration Record of the `linear` model calls `gain` (Section 8.12). A correction that equipment applies by itself as part of its calibration, such as a probe's, is not recorded here.
5. A consumer that shows a value with `unitAssumed` SHOULD show that its basis was assumed, and Section 9.4 says when a product may act on it.

A Method is an object with the following members. Its `type` says how the value was measured, and its `equipment` what measured it: a result of an automatic analyzer names the analyzer, so that consumers can tell it from a person's test of the same type, such as a titration (Table 25).

Table 8 — Method members

Member	Type	Presence	Description
type	string	REQUIRED	Method type (Table 25).
equipment	Identifier	OPTIONAL	The Equipment that produced the value, such as a sensor, an analyzer, or a controller.
instrument	object	OPTIONAL	An instrument or test kit that is not recorded as Equipment, with OPTIONAL <code>manufacturer</code> and <code>model</code> strings. At least one of the two MUST be present.
referenceTemperature	Quantity	OPTIONAL	For <code>specific-gravity</code> only: the temperature that the instrument's scale is referred to (see below). The unit MUST be a temperature unit.
referenceElectrode	string	OPTIONAL	For <code>orp</code> only: the reference electrode that the value is measured against (Table 27).

Member	Type	Presence	Description
extensions	Extensions	OPTIONAL	

Specific gravity. Specific gravity is the density of the water divided by the density of pure water, each at a stated temperature. Hydrometers and refractometers are referred to different temperatures, and the same water gives a different value at each. For example, seawater of salinity 35 has a specific gravity of 1.0269 referred to 15.56 °C (60 °F), 1.0266 referred to 20 °C, and 1.0264 referred to 25 °C, from the equation of state of seawater [UNESCO-44]. The difference, about 0.0005, is about 0.7 of salinity: near the resolution of hobby instruments, but enough to make two instruments disagree.

1. `referenceTemperature` is the temperature to which both the water and the pure water are referred, as the instrument's documentation states it, such as 20 °C for a refractometer whose scale reads specific gravity at 20/20 °C.
2. Producers SHOULD record `referenceTemperature` when the instrument states it.
3. Consumers SHOULD show it with the value. This Specification defines no conversion between reference temperatures, and none between specific gravity and salinity (Section 9.5).

ORP. ORP is the potential of an electrode in the water against a reference electrode. Readings against a silver chloride reference are about 200 mV lower than against the standard hydrogen electrode, and differ by about 10 mV between fillings, which Table 27 gives [USGS] [OAP-PI Section 8.5].

1. Producers SHOULD record `referenceElectrode` when the electrode's documentation names its reference.
2. Consumers SHOULD show it with the value. This Specification defines no conversion between reference electrodes.

8.5 Sample

Table 9 — Sample members

Member	Type	Presence	Description
id	Identifier	REQUIRED	
system	Identifier	REQUIRED	The system the water was taken from.
vessel	Identifier	OPTIONAL	The vessel the water was taken from. MUST belong to <code>system</code> .
collected	Timestamp	REQUIRED	When the water was taken.
analyzed	Timestamp	OPTIONAL	When the analysis was completed.
laboratory	object	OPTIONAL	The laboratory that analyzed the sample: <code>name</code> (string, REQUIRED) and <code>reference</code> (string, OPTIONAL; the laboratory's own sample or report number).
notes	string	OPTIONAL	Free text.
extensions	Extensions	OPTIONAL	

A laboratory that does not know the owner's system identifiers SHOULD include a System with the name the owner gave when registering the sample. Consumers that import such a Document SHOULD let the owner choose which existing system the results belong to.

8.6 Event

Table 10 — Event members

Member	Type	Presence	Description
id	Identifier	OPTIONAL	
type	string	REQUIRED	Event type (Table 26).
time	Timestamp	REQUIRED	When the event happened.
end	Timestamp	OPTIONAL	When the event ended, for events that last a period of time, such as a power outage. MUST NOT be earlier than <code>time</code> .
occurrences	integer	OPTIONAL	The number of occurrences the event totals (see below). At least 1. Default 1.
system	Identifier	REQUIRED	The system affected.

Member	Type	Presence	Description
vessel	Identifier	OPTIONAL	The vessel affected. MUST belong to <code>system</code> .
equipment	Identifier	OPTIONAL	The Equipment that performed or was the subject of the event.
channel	Identifier	OPTIONAL	The Channel of <code>equipment</code> involved. If present, <code>equipment</code> MUST be present and MUST have a Channel with this <code>id</code> .
amount	Quantity	OPTIONAL	The volume or mass involved. The unit MUST be a volume or mass unit.
wasted	Quantity	OPTIONAL	The water sent to the drain, such as an RO/DI system's concentrate and flushes. The unit MUST be a volume unit.
product	Product	OPTIONAL	The product used (see below).
calibration	Calibration Record	OPTIONAL	For <code>calibration</code> events only: what the calibration did (Section 8.12).
alarm	Alarm	see description	For <code>alarm</code> , <code>restart</code> , and <code>power-outage</code> events only: the alarm or notice (Section 8.13). REQUIRED for <code>alarm</code> .
livestock	Identifier	see description	For <code>livestock-added</code> and <code>livestock-removed</code> events only: the Livestock entry. REQUIRED for those types.
count	integer	OPTIONAL	For <code>livestock-added</code> and <code>livestock-removed</code> events only: how many individuals, colonies, or plants. At least 1.
reason	string	OPTIONAL	For <code>livestock-removed</code> events only: why (Table 32).
task	Identifier	OPTIONAL	The Task the event completes (Section 8.11).
element	Identifier	see description	For <code>mode</code> , <code>override</code> , and <code>program</code> events only: the element of the Automation configuration (Section 7.3, item 4). REQUIRED for those types.
notes	string	OPTIONAL	Free text. REQUIRED when <code>type</code> is <code>note</code> .
extensions	Extensions	OPTIONAL	

A Product is an object with `name` (string, REQUIRED), `manufacturer` (string, OPTIONAL), and `gtin` (string, OPTIONAL). `gtin` is the GS1 Global Trade Item Number printed under the product's barcode [GS1]: 8, 12, 13, or 14 digits, the last of which is a check digit. The check digit is the number that, added to the sum of the other digits weighted 3 and 1 alternately from the right, gives a multiple of 10. A GTIN identifies a product in one pack size, so the same product in another size has another GTIN.

Requirements:

1. A member that Table 10 gives "for" some event types MUST NOT appear in events of other types.
2. **Totals.** A producer SHOULD record each occurrence of an event as an event of its own. A producer that records many, such as a day's top-off fills or a doser's frequent small doses, MAY instead record the occurrences of one type, place, equipment, channel, and product within one day as one event. The day is a local day, or a day as the specification of the equipment defines it, such as a dosing day [OAP-DO Section 7.5] or a testing day [OAP-WT Section 8.3], which may cross midnight. So the event's `end` is never more than 25 hours after its `time`, the longest local day. That event has `occurrences`, the number of occurrences; `time`, the start of the first; `end`, the end of the last; and `amount`, their total. The specification for the kind of equipment says which occurrences may be totaled: missed doses, for example, are always recorded one by one [OAP-DO Section 16.1] [OAP-WL Section 9.15]. In Documents of revisions 0.1 and 0.2, the day is a local day.
3. **Calibrations.** The `time` of a `calibration` event SHOULD be when the calibration ended, from when its result applies.
4. **Automation.** A controller SHOULD record each period in which a mode was active, each override, and each run of a step program [OAP-AU Sections 14 to 16] as an event of the type `mode`, `override`, or `program`, so that owners can see why equipment did what it did. `element` is the `id` of the mode, the output, or the step program. Controller configuration itself is not part of a Document: it moves in the Automation specification's configuration documents [OAP-AU Section 19], and the alarm settings in the Alarms and Notifications specification's alarm settings documents [OAP-AN Section 19]. A controller that exports both SHOULD use the same identifiers for systems and equipment in each [OAP-AU Section 7.4].

Specifications for kinds of equipment say which events their functions record, and keep details that belong to them alone in the extensions of the events, under their own extension names (Section 7.6). Table 11 lists them.

Table 11 — Records that other specifications define

Record	Event type, or object	Details in extensions	Defined in
Doses, given, missed, or skipped	dose	org.openaquariumproject.dosing	[OAP-DO Section 16.1]
Planned dosing amounts	dose-adjusted	org.openaquariumproject.dosing	[OAP-DO Section 16.2]
Calibrations of dosing heads, and refills of their containers	calibration, refill	org.openaquariumproject.dosing	[OAP-DO Section 16.3]
Results of analyzers, manual tests, and laboratories	Observation	org.openaquariumproject.water-testing	[OAP-WT Section 15.1]
Tests of analyzers	analyzer-test	org.openaquariumproject.water-testing	[OAP-WT Section 15.2]
Comparisons, corrections, and analyzers' own calibrations	calibration	org.openaquariumproject.water-testing	[OAP-WT Section 15.3]
Reagents replaced and waste emptied	maintenance	org.openaquariumproject.water-testing	[OAP-WT Section 15.4]
Calibrations of probes	calibration	org.openaquariumproject.probe-calibration, in the extensions of the event's Calibration Record	[OAP-PI Section 14.9]
Water changes, top-off, new water, refills of reservoirs, and RO/DI runs	water-change, top-off, refill, ro-di-run, Observation	—	[OAP-WL Section 9.15]
Alarms, power outages, and restarts	alarm, power-outage, restart	—	[OAP-AN Section 20]
Measured light	ppfd, daily-light-integral	—	[OAP-LP Section 19]
Measured flow and turnover	flow-rate, turnover	—	[OAP-FP Section 14.3]

8.7 Series

A Series records many values of one parameter, measured in one place by one method, in compact form. It is meant for sensors that record values often, such as every minute, where one Observation for each value would be large.

Table 12 — Series members

Member	Type	Presence	Description
id	Identifier	OPTIONAL	
parameter	string	REQUIRED	Parameter identifier, as for an Observation.
unit	string	REQUIRED	Unit code of every value. MUST be accepted for parameter.
system	Identifier	REQUIRED	The system the values describe.
vessel	Identifier	OPTIONAL	The vessel the values describe, as for an Observation.
method	Method	OPTIONAL	How the values were obtained.
start	Timestamp	REQUIRED	The time from which interval or times count.
interval	number (s)	see below	Seconds between values. Greater than 0.
times	array of number (s)	see below	For each value, seconds after start.
values	array	REQUIRED	Numbers in unit, or null where there is no value. At least one.

Member	Type	Presence	Description
statistic	string	OPTIONAL	mean, min, or max: each value summarizes the readings over its interval (see below). This vocabulary is closed.
notes	string	OPTIONAL	Free text.
extensions	Extensions	OPTIONAL	

Requirements:

1. A Series MUST have exactly one of `interval` and `times`.
2. With `interval`, value i , counting from 0, applies at `start` plus $i \times \text{interval}$ seconds.
3. With `times`, `times` MUST have as many items as `values`, each at least 0 and greater than the one before it. Value i applies at `start` plus `times[i]` seconds.
4. A `null` value means that there is no value at that time, for example because the sensor's reading was not good.
5. A Series without `statistic` is equivalent to one Observation for each of its values that is not `null`, with the Series' `parameter`, `unit`, `system`, `vessel`, and `method`, and the value's time, expressed in the UTC offset of `start`. Consumers MAY convert between the two forms. A producer MUST NOT record the same value both ways.
6. With `statistic`, the Series MUST have `interval`, and value i is the mean, the smallest, or the largest of the readings from `start` plus $i \times \text{interval}$ seconds up to, but not including, `start` plus $(i + 1) \times \text{interval}$ seconds. Products use such Series to keep long histories small, for example as hourly means. Consumers MUST show such values as the statistic they are, and MUST NOT treat them as single readings.
7. Values in a Series are never qualified. A reading below or above a sensor's range is recorded as an Observation with its `qualifier` (Section 8.4).
8. A producer SHOULD start a new Series when the UTC offset where the system is located changes, such as at the start or end of daylight saving time, so that the offset of `start` stays the local one.
9. A Series SHOULD NOT hold more than 100 000 values (provisional). Producers SHOULD split long records, for example into one Series for each parameter, place, and local day.

For example, the temperature of a sump every 5 minutes for one hour, with one reading that was not good:

```
{ "parameter": "temperature", "unit": "degC", "system": "reef", "vessel": "reef-sump",
  "method": { "type": "sensor", "equipment": "temp-1" },
  "start": "2026-10-13T00:00:00-07:00", "interval": 300,
  "values": [25.42, 25.4, 25.38, 25.36, 25.35, 25.33, null, 25.36, 25.41, 25.45, 25.48, 25.5] }
```

8.8 State Series

A State Series records the states of one attribute of one piece of equipment over time, as its changes. It answers questions such as when an outlet was on, how long a heater heated each day, or when a float switch was last wet.

Table 13 — State Series members

Member	Type	Presence	Description
id	Identifier	OPTIONAL	
system	Identifier	REQUIRED	The system the equipment serves. MUST equal the equipment's <code>system</code> .
equipment	Identifier	REQUIRED	The Equipment whose states are recorded.
channel	Identifier	OPTIONAL	The Channel of <code>equipment</code> , such as one outlet of a power strip.
attribute	string	REQUIRED	The attribute recorded: a Device Classes attribute name [OAP-DC], such as <code>on</code> , <code>running</code> , <code>heating</code> , <code>wet</code> , or <code>leak</code> , or an extension name.
unit	string	see description	The unit of numeric values, such as % for a pump's <code>speed</code> . REQUIRED when the values are numbers, and absent otherwise.
start	Timestamp	REQUIRED	The time from which <code>times</code> count.
times	array of number (s)	REQUIRED	For each value, the seconds after <code>start</code> at which it began.

Member	Type	Presence	Description
values	array	REQUIRED	The states: booleans, numbers, or strings, as the attribute defines them, or <code>null</code> where the state was not known. At least one.
end	Timestamp	OPTIONAL	The time up to which the states are known. Not earlier than the time of the last value.
notes	string	OPTIONAL	Free text.
extensions	Extensions	OPTIONAL	

Requirements:

1. `times` MUST have as many items as `values`, each at least 0 and greater than the one before it.
2. Each value holds from its time until the time of the next value. The last value holds until `end`; without `end`, how long it held is not known.
3. A `null` value means that the state was not known, for example while the controller could not reach the device.
4. Apart from `null`, the values of one State Series MUST all be booleans, all numbers, or all strings.
5. A producer SHOULD record a value only when the state changes.
6. A State Series that records whether an output runs SHOULD record the attribute that the Automation specification uses to tell whether the output is running, such as `on` for a switch or `running` for a pump [OAP-AU Section 8.3], as the device reports it rather than as it was commanded. Other State Series record whatever attribute they are about, such as a heater's `heating` or a pump's `speed`.
7. Items 8 and 9 of Section 8.7 apply to State Series as well.

For example, a heater that heated from 00:20 to 00:45 and from 00:50 until at least 01:00:

```
{ "system": "reef", "equipment": "heater-1", "attribute": "heating",
  "start": "2026-10-13T00:00:00-07:00", "times": [0, 1200, 2700, 3000],
  "values": [false, true, false, true], "end": "2026-10-13T01:00:00-07:00" }
```

8.9 Range

A Range holds an owner's desired and acceptable values of one parameter, and the limits beyond which its values are dangerous, for one system or one vessel. Products draw ranges on charts and in reports, and carry them to a new product. The Alarms and Notifications specification raises alarms from them [OAP-AN Section 9].

Table 14 — Range members

Member	Type	Presence	Description
id	Identifier	REQUIRED	
name	string	OPTIONAL	Name shown to the owner.
system	Identifier	REQUIRED	The system.
vessel	Identifier	OPTIONAL	A vessel of <code>system</code> . Without it, the range applies to the system's circulating water.
parameter	string	REQUIRED	Parameter identifier, such as <code>temperature</code> or <code>nitrate</code> .
unit	string	REQUIRED	The unit of every number of the range. MUST be accepted for <code>parameter</code> .
desired	object	OPTIONAL	<code>min</code> , <code>max</code> , or both: the values the owner aims for.
acceptable	object	OPTIONAL	<code>min</code> , <code>max</code> , or both: the values the owner can accept for a while.
danger	object	OPTIONAL	<code>below</code> , <code>above</code> , or both: the limits beyond which a value is dangerous.
methods	array of string	OPTIONAL	Method types (Table 25). Only values measured by these methods count. Default: all.
notes	string	OPTIONAL	Free text.
extensions	Extensions	OPTIONAL	

Requirements:

1. A Range MUST have at least one of `desired`, `acceptable`, and `danger`.

- The bounds MUST be in order, for those that are given: `danger.below ≤ acceptable.min ≤ desired.min < desired.max ≤ acceptable.max ≤ danger.above`.
- A system or vessel MUST NOT have two Ranges for the same parameter.
- A Range with a `vessel` applies to values of that vessel. A Range without one applies to values of the system that have no vessel or a circulating vessel.
- To compare a value with a Range, a consumer converts the value to the Range's `unit` (Section 9.5). A qualified value counts as beyond a bound only where it must be: a result of `< 0.02` is below a `min` of 0.03, but a result of `> 0.5` is not known to be above an `above` of 1.0 [OAP-AN Section 9.3].
- Ranges are the owner's statement of what is right for their livestock. This Specification gives no default values, and a producer MUST NOT record a Range that the owner has not set or confirmed.
- The Alarms and Notifications specification adds `delay`, `hysteresis`, and `trend` to the ranges of its alarm settings documents [OAP-AN Section 9.1]. A Range in a Document has none of them. A product that keeps a range in both SHOULD give it the same `id` and bounds in each.

For example, a reef owner's temperature range might be desired 25.0 to 26.0 °C, acceptable 24.0 to 27.0 °C, and danger below 22.0 or above 29.0 °C:

```
{ "id": "range-temperature", "system": "reef", "parameter": "temperature", "unit": "degC",
  "desired": { "min": 25.0, "max": 26.0 }, "acceptable": { "min": 24.0, "max": 27.0 },
  "danger": { "below": 22.0, "above": 29.0 } }
```

8.10 Livestock

A Livestock entry describes organisms of one kind kept together in a system, such as a school of eight neon tetras, a pair of clownfish, or one coral colony.

Table 15 — Livestock members

Member	Type	Presence	Description
<code>id</code>	Identifier	REQUIRED	
<code>system</code>	Identifier	REQUIRED	The system where they are kept.
<code>vessel</code>	Identifier	OPTIONAL	The vessel where they are kept, such as a refugium. MUST belong to <code>system</code> .
<code>kind</code>	string	REQUIRED	Livestock kind (Table 30).
<code>taxon</code>	object	see below	The taxon they belong to: <code>scientificName</code> and <code>identifiers</code> (see below).
<code>commonName</code>	string	see below	A common or trade name, such as "Ocellaris clownfish".
<code>variety</code>	string	OPTIONAL	A variety, morph, color strain, or cultivar that taxonomy does not name, such as a named coral morph or a color strain of shrimp.
<code>name</code>	string	OPTIONAL	A name the owner gives them, such as "Nemo".
<code>count</code>	integer	OPTIONAL	How many individuals, colonies, or plants are kept now. 0 after all have left. Absent for organisms that are not counted, such as macroalgae.
<code>origin</code>	string	OPTIONAL	Where they came from (Table 31).
<code>added</code>	Date	OPTIONAL	When the first was added.
<code>removed</code>	Date	OPTIONAL	When the last left. Not earlier than <code>added</code> .
<code>notes</code>	string	OPTIONAL	Free text.
<code>extensions</code>	Extensions	OPTIONAL	

Requirements:

- A Livestock entry MUST have `taxon`, `commonName`, or both, and SHOULD have `taxon`.
- `taxon.scientificName` (string, REQUIRED) is the scientific name of the lowest taxon to which the organisms are identified, without its authorship, such as *Amphiprion ocellaris*, or, for a coral identified only to its genus, *Acropora*.
- `taxon.identifiers` (object, OPTIONAL) holds identifiers of that taxon in taxonomic databases. Each member name is a taxon scheme (Table 33) or an extension name, and each value is the identifier as a string, such as `{ "worms": "278400" }`.
- Producers SHOULD give at least one identifier, and consumers SHOULD match taxa by identifier rather than by name. Names are ambiguous and change: the World Register of Marine Species lists two genera

named *Acropora*, the accepted genus of stony corals (AphiaID 205469) and an unaccepted name (AphiaID 578625) [WoRMS], and when a species is renamed, its identifier still leads to its accepted name.

5. An entry with `removed` has a `count` of 0, if it has a `count`.
6. Events of the types `livestock-added` and `livestock-removed` record each change, with `count` the number added or removed. When a Document holds every such event of an entry, the entry's `count` SHOULD be the sum of those added less the sum of those removed.
7. Livestock moved to another system is recorded as removed from one, with the reason `moved`, and added to the other, as an entry of that system.

For example:

```
{ "id": "clownfish", "system": "reef", "kind": "fish",
  "taxon": { "scientificName": "Amphiprion ocellaris", "identifiers": { "worms": "278400", "gbif": "5212106" } },
  "commonName": "Ocellaris clownfish", "count": 2, "origin": "captive-bred", "added": "2024-07-02" }
```

8.11 Task

A Task is a piece of maintenance that people do by hand, once or on a schedule, such as cleaning a skimmer cup every week, replacing carbon every month, replacing a pump's impeller once, or replacing a UV lamp after 9000 hours of running. Products show which tasks are due, and remind people of them.

Table 16 — Task members

Member	Type	Presence	Description
<code>id</code>	Identifier	REQUIRED	
<code>name</code>	string	REQUIRED	What to do, as shown to the owner, such as "Clean the skimmer cup".
<code>system</code>	Identifier	REQUIRED	The system.
<code>vessel</code>	Identifier	OPTIONAL	The vessel. MUST belong to <code>system</code> .
<code>equipment</code>	Identifier	OPTIONAL	The Equipment the task is done on. MUST serve <code>system</code> .
<code>channel</code>	Identifier	OPTIONAL	The Channel of <code>equipment</code> . If present, <code>equipment</code> MUST be present.
<code>eventType</code>	string	OPTIONAL	The event type (Table 26) that records the task done. Default <code>maintenance</code> .
<code>every</code>	Recurrence	OPTIONAL	How often the task repeats. Absent for a task done once.
<code>everyHours</code>	number (h)	OPTIONAL	How many hours of running of <code>equipment</code> the task repeats after (see below). Greater than 0. Added in 0.3.
<code>fixed</code>	boolean	OPTIONAL	<code>true</code> when the due dates follow a calendar from <code>start</code> , whenever the task is done; <code>false</code> when each due date counts from when the task was last done. Default <code>false</code> .
<code>start</code>	Date	OPTIONAL	The first due date. REQUIRED when <code>fixed</code> is <code>true</code> .
<code>due</code>	Date	OPTIONAL	The next due date, as the producer last computed it (see below).
<code>remind</code>	string	OPTIONAL	How long before the due date to remind people, written as a Recurrence in days or weeks, such as <code>P2D</code> . Default: on the due date.
<code>ended</code>	Date	OPTIONAL	From when the task is no longer done.
<code>notes</code>	string	OPTIONAL	Free text.
<code>extensions</code>	Extensions	OPTIONAL	

Requirements:

1. An event completes a task when its `task` is the task's `id`. It SHOULD be of the task's `eventType`, and SHOULD have the task's `vessel`, `equipment`, and `channel`.
2. A task without `every` or `everyHours` is due once: on `due`, or on `start`. It is not due again once an event has completed it.
3. A task with `every` is due again after each time it is done:
 - when `fixed` is `false`, on the local date of the latest event that completed it plus `every`, or, before it is first done, on `start`, if given; and
 - when `fixed` is `true`, on the first of the dates `start`, `start` plus `every`, `start` plus $2 \times \text{every}$, and so on, each computed from `start`, that is later than the local date of the latest event that completed it, or, before it is first done, on `start`.

4. A task with `everyHours` MUST have `equipment`, and is due again once that equipment has run for `everyHours` hours since the latest event that completed it, or, before it is first done, since `start`, if given, or else since the equipment was `installed`. The hours the equipment has run are counted by the `operatingHours` of the Device Classes outputs that drive it [OAP-DC Section 8.3]: the most that any one of them has increased. Device Classes makes `operatingHours` OPTIONAL, so a product whose outputs do not report it MAY count the time they reported the equipment running, by the attribute of Section 8.8, item 6, and otherwise SHOULD tell the owner that it cannot tell when the task falls due. With both `every` and `everyHours`, the task is due at whichever comes first. Running time suits maintenance of what wears as it runs, such as a UV lamp rated for a number of hours, or a dosing head's tubing, which one maker advises calibrating every 100 hours of running [OAP-DO Appendix D, issue 22].
5. `due` is the next due date as the producer last computed it, including any postponement that a person made. For a task with `everyHours`, it is the date on which the producer expects the hours to be reached, from the equipment's recent running, and the producer SHOULD keep it current, so that a consumer that cannot count the hours still shows when the task falls due. A consumer that imports a Task SHOULD use `due`, rather than compute it again, until the task is next done.
6. A task with `ended` is never due. Producers SHOULD keep ended tasks, so that earlier events can still refer to them.
7. How a product reminds people is not defined by this Specification: the Alarms and Notifications specification reminds them of tasks that fall due with its notice `task-due` [OAP-AN].

For example, a task done every week, counted from when it was last done, and one done after every 9000 hours of running:

```
{ "id": "task-skimmer-cup", "name": "Clean the skimmer cup", "system": "reef", "equipment": "skimmer-1",
  "every": "P7D", "due": "2026-10-17" }
```

```
{ "id": "task-uv-lamp", "name": "Replace the UV lamp", "system": "reef", "equipment": "uv-1",
  "everyHours": 9000, "due": "2027-10-29" }
```

8.12 Calibration Record

A Calibration Record says what a calibration did: the references used, what the equipment read at each, the result, and the result it replaced. It is the `calibration` member of a `calibration` event, whose `equipment` and `channel` identify what was calibrated, and whose `time` is when the calibration ended. A Calibration Record that no event holds, such as a device's `lastCalibration` [OAP-DC] or the record that a smart probe stores [OAP-PI Section 14.9], gives that time as its own `time`.

Table 17 — Calibration record members

Member	Type	Presence	Description
<code>outcome</code>	string	REQUIRED	<code>completed</code> : the result was applied; <code>checked</code> : the equipment was compared with references and not adjusted; <code>failed</code> : the result was refused; or <code>cancelled</code> . This vocabulary is closed.
<code>reason</code>	string	OPTIONAL	Why it failed or was cancelled (Table 29). SHOULD be present for <code>failed</code> and <code>cancelled</code> , and MUST NOT be present otherwise.
<code>parameter</code>	string	OPTIONAL	The parameter that the equipment measures or delivers, such as <code>ph</code> or <code>flow-rate</code> .
<code>model</code>	string	OPTIONAL	How the result is applied (Table 28).
<code>points</code>	array of Calibration Point	OPTIONAL	The references, in the order they were used.
<code>result</code>	Calibration Result	see description	What the calibration computed: for <code>completed</code> , the result now in use; for <code>failed</code> , the result that was refused, if one was computed. REQUIRED for <code>completed</code> , unless the equipment does not report what its calibration computed (item 5); MUST NOT be present for <code>checked</code> and <code>cancelled</code> .
<code>previous</code>	Calibration Result	OPTIONAL	The result in use before the calibration. After a <code>checked</code> , <code>failed</code> , or <code>cancelled</code> calibration, it is still in use.

Member	Type	Presence	Description
time	Timestamp	OPTIONAL	When the calibration ended, for a record that no event holds. <i>Added in 0.4.</i>
due	Timestamp	OPTIONAL	When the next calibration falls due.
extensions	Extensions	OPTIONAL	

Table 18 — Calibration point members

Member	Type	Presence	Description
reference	Quantity	REQUIRED	The true value at the point, such as a buffer's pH at its temperature, a laboratory result, or the flow a dosing head was measured to deliver. If the record has a <code>parameter</code> , the unit MUST be accepted for it.
reading	Quantity	OPTIONAL	What the equipment read at the point before the calibration: a probe's raw signal, such as -3.2 mV, an analyzer's uncorrected result, or the flow a dosing head was asked for.
temperature	Quantity	OPTIONAL	The temperature of the reference. The unit MUST be a temperature unit.
solution	Product	OPTIONAL	The reference solution, such as a pH 7.00 buffer.
time	Timestamp	OPTIONAL	When the point was taken.
extensions	Extensions	OPTIONAL	

A Calibration Result is an object with at least one of these members:

- `offset` (Quantity): for `nernst`, the electrode's potential at pH 7, in an electric potential unit; for `linear`, the amount added, in the unit of the readings;
- `slope` (Quantity): for `nernst`, the electrode's slope, in % of the theoretical slope;
- `gain` (number, greater than 0): for `linear`, the factor that readings are multiplied by, which an observation's `correction` calls `factor` (Section 8.4);
- `rate` (Quantity): for `flow-rate`, the flow delivered, in a volumetric flow unit; and
- `extensions` (Extensions).

Requirements:

1. The specification for a kind of equipment defines its calibration procedure and its models: Probe Interface for probes [OAP-PI Section 14], Dosing for dosing heads [OAP-DO Section 8], and Water Testing for comparisons and corrections of analyzers [OAP-WT Section 11]. A Calibration Record holds what they have in common. Details that belong to one kind of equipment, such as a conductivity probe's compensation method, are kept in extensions (Table 11).
2. A product SHOULD record every calibration, whatever its outcome, and SHOULD keep the records for the life of the equipment, so that the trends of slope and offset show its condition [OAP-PI Section 14.10].
3. The age of a probe is the time since its Equipment's `installed` date, and its expected end is the Equipment's `endOfLife` (Section 8.3).
4. A correction of an analyzer's results is a Calibration Record of the `linear` model whose `gain` is the correction's `factor`, and whose `offset` is its `offset`, in the unit of the results [OAP-WT Section 15.3]. The observations it corrects carry the same `factor` and `offset` in their `correction` (Section 8.4).
5. Some equipment adjusts itself without reporting what its calibration computed, such as an analyzer that calibrates its own pumps against a standard [OAP-WT Section 11.8]. A `completed` Calibration Record of such equipment leaves out `result`, and says only that the equipment calibrated itself; the result in use is then not known.
6. A Calibration Record that an event holds MUST NOT have a `time` other than the event's `time`.

For example, a two-point calibration of a pH electrode:

```
{ "type": "calibration", "time": "2026-10-01T18:30:00-07:00", "system": "reef", "equipment": "ph-1",
  "calibration": {
    "outcome": "completed", "parameter": "ph", "model": "nernst",
    "points": [
      { "reference": { "value": 7.0, "unit": "pH" }, "reading": { "value": -3.2, "unit": "mV" }, "temperature": { "value": 24.6,
        "unit": "degC" } },
      { "reference": { "value": 10.0, "unit": "pH" }, "reading": { "value": -172.1, "unit": "mV" }, "temperature": { "value":
        24.6, "unit": "degC" } } ],
    "result": { "offset": { "value": -3.2, "unit": "mV" }, "slope": { "value": 95.3, "unit": "%" } },
    "previous": { "offset": { "value": -1.8, "unit": "mV" }, "slope": { "value": 97.1, "unit": "%" } },
    "due": "2027-01-01T18:30:00-08:00" } }
```

8.13 Alarm

An Alarm is an alarm or a notice as the Alarms and Notifications specification defines it, with the members of its Table 25 [OAP-AN]. `id`, `code`, `kind`, `severity`, `title`, `raised`, `active`, and `subject` are REQUIRED, and `notified` gives how it was notified, as kept in history [OAP-AN Section 20.1]. This Specification does not repeat that definition.

1. The `time` of an `alarm` event MUST equal its alarm's `raised`, and its `end`, if both are present, its alarm's `cleared`.
2. The `system`, `vessel`, and `equipment` of the event SHOULD be those of the alarm. Those of the alarm MUST be present in the Document, and its other identifiers need not be (Section 7.3, item 5).
3. An alarm about a controller as a whole is recorded once for each system the controller serves [OAP-AN Section 20.1], with the same `id` each time (Section 7.3, item 1).
4. A `power-outage` notice is held by the `power-outage` event of the outage, and a `controller-restarted` or `device-restarted` notice by a `restart` event [OAP-AN Section 20.2].
5. Alarms carry the names of the people who acknowledged them (Section 11).

9 Registry

9.1 General

The registry defines the parameters, units, unit conversions, and vocabularies used in Documents. Its content is published in machine-readable form in `registry.json`, from which the tables in this section are generated. `registry.json` is normative: where it and this document differ, `registry.json` governs, and the difference is an error in this document to be corrected in the next revision.

Each vocabulary in `registry.json` is a list of entries, each with an `id` and a `description`. In revision 0.1, vocabularies were lists of names, and their tables were written by hand. Parameters, units, and vocabulary entries that a revision after 0.1 added have `since`, the revision that added them, and the tables mark them.

Registry names (parameter identifiers and vocabulary values defined here) consist of lowercase letters, digits, and single hyphens, and never contain a dot. This keeps them distinct from extension names.

9.2 Parameters

Table 19 lists the parameters defined by this revision. For each parameter, an observation's `unit` MUST be the canonical unit or one of the other accepted units. The factor converts a value in that unit to the canonical unit (Section 9.5).

Table 19 — Parameters

Identifier	Name	Canonical unit	Other accepted units (factor)	Notes
<code>temperature</code>	Temperature	<code>degC</code>	<code>degF</code> ($\times 0.5555555556$, then -17.777777778)	
<code>salinity</code>	Salinity	<code>psu</code>	<code>ppt</code> ($\times 1$)	<code>ppt</code> is treated as equal to <code>psu</code> ; the difference between practical and absolute salinity (about 0.5 ‰) is below hobby measurement precision.
<code>specific-gravity</code>	Specific gravity	<code>SG</code>	—	Not convertible to salinity without temperature; recorded as its own parameter.
<code>conductivity</code>	Conductivity	<code>mS/cm</code>	<code>uS/cm</code> ($\times 0.001$)	Specific conductance, temperature-compensated to 25 °C.
<code>ph</code>	pH	<code>pH</code>	—	

Identifier	Name	Canonical unit	Other accepted units (factor)	Notes
orp	Oxidation-reduction potential	mV	—	
alkalinity	Alkalinity	meq/L	dKH ($\times 0.356652$), mg/L-CaCO ₃ ($\times 0.0199828$)	Total alkalinity. Freshwater "KH" results are alkalinity and use this parameter.
general-hardness	General hardness	mmol/L	dGH ($\times 0.178326$), mg/L-CaCO ₃ ($\times 0.00999141$)	Calcium plus magnesium.
calcium	Calcium	mg/L	—	Element Ca.
magnesium	Magnesium	mg/L	—	Element Mg.
potassium	Potassium	mg/L	—	Element K.
strontium	Strontium	mg/L	ug/L ($\times 0.001$)	Element Sr.
iodine	Iodine	mg/L	ug/L ($\times 0.001$)	Element I. Total iodine (iodide, iodate, and organic forms).
iron	Iron	mg/L	ug/L ($\times 0.001$)	Element Fe.
copper	Copper	mg/L	ug/L ($\times 0.001$)	Element Cu.
nitrate	Nitrate	mg/L-N ₀₃	mg/L-N ($\times 4.42664$)	
nitrite	Nitrite	mg/L-N ₀₂	mg/L-N ($\times 3.28443$)	
ammonia	Ammonia (total)	mg/L-N	mg/L-NH ₃ ($\times 0.822441$), mg/L-NH ₄ ($\times 0.776484$)	Total ammonia (NH ₃ + NH ₄ ⁺).
free-ammonia	Ammonia (free)	mg/L-NH ₃	mg/L-N ($\times 1.21589$)	Un-ionized ammonia (NH ₃) only, as free-ammonia indicators measure. Not convertible to ammonia without pH, temperature, and salinity. <i>Added in 0.2.</i>
phosphate	Phosphate	mg/L-P ₀₄	mg/L-P ($\times 3.06612$), ug/L-P ($\times 0.00306612$)	Reactive (ortho)phosphate, as measured by colorimetric tests. Total phosphorus from elemental analysis is <code>element:P</code> .
silicate	Silicate	mg/L-Si ₀₂	mg/L-Si ($\times 2.13933$)	Reactive silicate. Total silicon from elemental analysis is <code>element:Si</code> .
carbon-dioxide	Carbon dioxide (dissolved)	mg/L	—	
dissolved-oxygen	Dissolved oxygen	mg/L	—	
total-dissolved-solids	Total dissolved solids	mg/L	—	Usually estimated from conductivity by the instrument.
ppfd	Photosynthetic photon flux density	umol/m ² /s	—	PAR at a point. Describe the measurement position in <code>notes</code> .
daily-light-integral	Daily light integral	mol/m ² /d	—	Photons received at a point over one local day; an observation's <code>time</code> is the start of that day. Describe the measurement position in <code>notes</code> . <i>Added in 0.2.</i>
flow-rate	Flow rate	L/h	L/min ($\times 60$), gal/h ($\times 3.785411784$), mL/min ($\times 0.06$)	
turnover	Turnover	1/h	—	How many times an hour pumps move a volume of water equal to the vessel's, or, without a vessel, to the system's water volume. <i>Added in 0.2.</i>
dissolved-oxygen-saturation	Dissolved oxygen saturation	%	—	Not convertible to <code>dissolved-oxygen</code> without temperature and salinity.

Identifier	Name	Canonical unit	Other accepted units (factor)	Notes
air-temperature	Air temperature	degC	degF ($\times 0.5555555556$, then -17.7777777778)	Air temperature near the system.
relative-humidity	Relative humidity	%	—	Relative humidity of the air near the system.
water-level	Water level	cm	mm ($\times 0.1$), in ($\times 2.54$)	Height of the water surface above a reference point chosen by the producer, such as the bottom of the sensor.
pump-speed	Pump speed	rpm	—	Rotational speed of a pump motor.
voltage	Voltage	V	mV ($\times 0.001$)	Electrical supply voltage, for example at an outlet.
current	Current	A	—	Electrical current drawn, for example by the equipment on an outlet.
power	Power	W	—	Real electrical power drawn.
energy	Energy	kWh	Wh ($\times 0.001$)	Electrical energy used since a counter was reset. Use over a period is the difference between two values.

Conversion factors are derived from IUPAC standard atomic weights [IUPAC] and rounded to six significant figures.

Most ammonia test kits measure total ammonia, NH_3 and NH_4^+ together, which is **ammonia**. Free-ammonia indicators measure un-ionized ammonia, NH_3 alone, which is **free-ammonia**. The share of total ammonia that is free depends on pH, temperature, and salinity, so the two are different parameters.

An observation of **daily-light-integral** covers one local day, and its **time** is the start of that day [OAP-LP Section 19].

9.3 Elements

The total concentration of a chemical element, typically measured by ICP-OES or ICP-MS, uses the parameter identifier **element**: followed by the element's IUPAC symbol with its standard capitalization, such as **element:Li** or **element:Mo**.

1. The symbol MUST be one of the 118 element symbols listed in **registry.json**.
2. For an element that has a named parameter in Table 19 (Ca, Mg, K, Sr, I, Fe, and Cu), producers MUST use the named parameter and MUST NOT use **element**.
3. Element parameters have canonical unit **mg/L** and also accept **ug/L** ($\times 0.001$). Values are expressed as the element.

Phosphorus and silicon have no named element parameter. A colorimetric phosphate or silicate test measures the reactive form and uses **phosphate** or **silicate**, while elemental analysis measures the total and uses **element:P** or **element:Si**. Consumers MUST NOT treat these as the same parameter.

9.4 Units

Unit codes are case-sensitive ASCII strings. **u** stands for the prefix micro (μ). A suffix of the form **-X** on a concentration unit gives the chemical basis: the value is the mass of species X per volume of water. For example, **mg/L-N** means milligrams of nitrogen per litre.

Unit codes MUST be taken from Table 20, except where Section 9.6 permits an extension unit. In particular, producers MUST NOT use "ppm" or "ppb". Many instruments label results as "ppm" or "ppb" without stating whether they mean mass per volume or mass per mass, or which chemical species the value is expressed as. A

producer reading such an instrument MUST record the value in the corresponding unit from Table 20 (normally mg/L or ug/L with the basis stated by the instrument's documentation).

Table 20 — Units

Code	Meaning	Factor to the base unit
degC	Degree Celsius	—
degF	Degree Fahrenheit	—
psu	Practical salinity (PSS-78), dimensionless	—
ppt	Parts per thousand by mass (g/kg); for salinity only	—
SG	Specific gravity (relative density), dimensionless	—
pH	pH, dimensionless	—
mV	Millivolt	—
mS/cm	Millisiemens per centimetre	—
uS/cm	Microsiemens per centimetre	—
meq/L	Milliequivalents per litre	—
mmol/L	Millimoles per litre	—
dKH	Degree of carbonate hardness (German); 1 dKH is equivalent to 10 mg/L CaO	—
dGH	Degree of general hardness (German); 1 dGH is equivalent to 10 mg/L CaO	—
mg/L -CaCO3	Milligrams per litre, expressed as calcium carbonate	—
mg/L	Milligrams per litre, expressed as the parameter's own species or element	—
ug/L	Micrograms per litre, expressed as the parameter's own species or element	—
mg/L -NO3	Milligrams per litre, expressed as nitrate (NO3-)	—
mg/L -NO2	Milligrams per litre, expressed as nitrite (NO2-)	—
mg/L -N	Milligrams per litre, expressed as nitrogen (N)	—
mg/L -NH3	Milligrams per litre, expressed as ammonia (NH3)	—
mg/L -NH4	Milligrams per litre, expressed as ammonium (NH4+)	—
mg/L -PO4	Milligrams per litre, expressed as phosphate (PO4 3-)	—
mg/L -P	Milligrams per litre, expressed as phosphorus (P)	—
ug/L -P	Micrograms per litre, expressed as phosphorus (P)	—
mg/L -SiO2	Milligrams per litre, expressed as silica (SiO2)	—
mg/L -Si	Milligrams per litre, expressed as silicon (Si)	—
umol/m2/s	Micromoles of photons per square metre per second	—
umol/s	Micromoles of photons per second, such as the photon flux of a light's channel. <i>Added in 0.3.</i>	—
mol/m2/d	Moles of photons per square metre per day. <i>Added in 0.2.</i>	—
L/h	Litres per hour	—
L/min	Litres per minute	—
mL/min	Millilitres per minute	—
gal/h	US gallons per hour	—
L	Litre	1 L
mL	Millilitre	0.001 L
gal	US gallon (3.785411784 L)	3.785411784 L
gal-imp	Imperial gallon (4.54609 L)	4.54609 L
g	Gram	1 g
mg	Milligram	0.001 g
%	Percent	—
%/degC	Percent per degree Celsius, such as the temperature coefficient of a conductivity. <i>Added in 0.3.</i>	—
s	Second	—

Code	Meaning	Factor to the base unit
h	Hour	—
V	Volt	—
A	Ampere	—
W	Watt	—
Wh	Watt-hour	—
kWh	Kilowatt-hour	—
mm	Millimetre	0.001 m
cm	Centimetre	0.01 m
m	Metre. <i>Added in 0.2.</i>	1 m
in	Inch (2.54 cm)	0.0254 m
nm	Nanometre	0.000000001 m
rpm	Revolutions per minute	—
Hz	Hertz	—
1/h	Per hour, such as volumes of water per hour. <i>Added in 0.2.</i>	—
K	Kelvin; used for the correlated color temperature of light	—

Volume units (L, mL, gal, gal-imp), mass units (g, mg), and length units (mm, cm, m, in, nm) are used in Quantities, and Table 20 gives each its factor to the base unit of its kind (Section 7.5). Units such as s, %, %/degC, K, and Hz are used by the attributes defined in the Device Classes specification [OAP-DC], and $\mu\text{mol/s}$ gives the photon flux of a source of light, such as one channel of a light. Their kinds are given in `registry.json`.

Results with no stated basis. Some test kits do not state the basis of their results: whether a nitrate result is nitrate or nitrogen, or whether an alkalinity result in “ppm” is calcium carbonate. The bases differ by up to 4.4 times (Section 2.4).

1. A producer MUST use the basis that the method’s documentation, label, color chart, or maker states.
2. When it is not stated, a producer that records a result that a person enters SHOULD ask the person, showing what each basis would mean, for example that 10 mg/L as N is 44 mg/L as NO₃.
3. When the basis is still not known, the producer MUST record the result in the unit of Table 21 (provisional), and MUST set the observation’s `unitAssumed` to `true`.
4. A product MUST NOT act on a value whose `unitAssumed` is `true` until a person has confirmed its basis: it does not dose from it [OAP-DO Section 12], raise or clear an alarm from it, or correct an instrument from it [OAP-WT Section 11]. It MAY show the value, with its basis marked as assumed (Section 8.4, item 5).
5. A person confirms a basis by choosing it, after the product has shown what each basis would mean, as item 2 says. The product then records the observation in the unit of the basis chosen, with the same `value`, and without `unitAssumed`. The confirmation needs no member of its own: the observation is then one whose basis is known.

Table 21 — Units assumed when a method states no basis (provisional)

Parameter	Assumed unit	Why
nitrate	mg/L-N03	The species the result is named for.
nitrite	mg/L-N02	The species the result is named for.
ammonia	mg/L-NH3	Kits that name both NH ₃ and NH ₄ ⁺ are assumed to report as NH ₃ ; those that show a basis report as NH ₃ or label their results NH ₄ , which differ by 6 %.
phosphate	mg/L-P04	The species the result is named for.
silicate	mg/L-Si02	The species the result is named for.
alkalinity, general-hardness	mg/L-CaCO3	For results in “ppm”; results in degrees are dKH or dGH.

9.5 Unit Conversion

A value in an accepted unit converts to the parameter’s canonical unit as:

canonical value = value × factor + offset

The offset is zero for every unit except `degF`. For example, 8.3 dKH of alkalinity is $8.3 \times 0.356652 = 2.96$ meq/L, and 1.5 mg/L-N of nitrate is $1.5 \times 4.42664 = 6.64$ mg/L-NO₃.

1. Consumers that convert values MUST use the factors in Table 19.
2. Conversion applies only between units of the same parameter. Consumers MUST NOT convert between parameters, such as specific gravity and salinity, as a unit conversion.
3. Consumers SHOULD display a converted value rounded to the original's resolution, converted. The resolution of a value is one unit of its last written digit, such as 0.1 for 8.3 or 1 for 420; converted with the factor alone, as item 5 says, its first significant digit gives the last decimal place of the converted value. For example, 0.1 dKH is 0.0357 meq/L, so 8.3 dKH is shown as 2.96 meq/L; and 0.1 meq/L is 0.28 dKH, so 3.0 meq/L is shown as 8.4 dKH. Counting significant digits instead would show 8.3 dKH as 3.0 meq/L, which reads back as 8.4 dKH.
4. Producers SHOULD record values in the unit the instrument or laboratory reported, and SHOULD NOT convert them before writing.
5. A difference between two values, such as a correction's `offset`, converts with the factor alone, without the offset.

9.6 Extension Parameters and Units

A producer MAY record a parameter that the registry does not define.

1. The parameter identifier MUST be an extension name (Section 7.6), such as `com.example.coral-color-index`.
2. Every extension parameter used in a Document MUST be defined in the top-level `parameters` array with a Parameter Definition: `id` (extension name, REQUIRED), `name` (string, REQUIRED), `units` (array of unit codes, REQUIRED, at least one), `description` (string, OPTIONAL), and `extensions` (OPTIONAL).
3. Observations and Series of an extension parameter MUST use one of the units listed in its definition.
4. An extension parameter SHOULD use units from Table 20. Where none is suitable, a unit code MAY be an extension name; its meaning SHOULD be explained in the definition's `description`.

Consumers cannot convert extension parameters, because the registry defines no factors for them. A producer that needs a parameter likely to be useful to others is encouraged to propose it for the registry.

9.7 Vocabularies

The vocabularies below are open: a value is either one defined here or an extension name. The closed vocabularies are `water` (Section 8.1), the `statistic` of a Series (Section 8.7), and the `outcome` of a Calibration Record (Section 8.12).

Table 22 — System tags (informative)

Tag	Meaning
<code>reef</code>	Marine system kept mainly for corals and other invertebrates.
<code>fish-only</code>	Kept mainly for fish.
<code>planted</code>	Kept mainly for aquatic plants.
<code>pond</code>	Outdoor or indoor pond.
<code>biotope</code>	Recreates a specific natural habitat.
<code>nano</code>	Small system.
<code>quarantine</code>	Holds new or sick livestock apart from the main system.
<code>breeding</code>	Used to breed or raise livestock.
<code>public</code>	Public display, such as a public aquarium or store display.
<code>research</code>	Used for research or education.

Table 23 — Vessel roles

Role	Meaning
display	The main vessel where livestock is kept and viewed, including a pond.
sump	A vessel, usually below the display, that holds equipment and adds water volume.
refugium	A vessel or compartment for growing macroalgae or plants, or for sheltering small organisms.
frag	A vessel for growing coral fragments or plant cuttings that shares the system's water.
reservoir	Holds water for the system, such as top-off or new water. Not circulating unless <code>circulating</code> says so (Section 8.2).
source	A supply of water that the system does not store, such as tap water, well water, or the product of an RO/DI unit on its way to a reservoir. Never circulating. <i>Added in 0.2.</i>
other	Any other role.

Table 24 — Equipment categories

Category	Meaning
controller	Monitors sensors and controls other equipment.
heater	Heats water.
chiller	Cools water.
return-pump	Returns water from a sump or filter to the display.
circulation-pump	Moves water within a vessel, such as a powerhead or wavemaker.
light	Lighting.
doser	Adds measured amounts of liquid, such as a peristaltic dosing pump.
auto-top-off	Replaces water lost to evaporation.
protein-skimmer	Removes dissolved organics by foam fractionation.
filter	Mechanical, chemical, or biological filtration, such as a filter sock, canister, or roller filter.
reactor	Media, calcium, kalkwasser, or other reactor.
uv-sterilizer	Ultraviolet sterilizer or clarifier.
ozone-generator	Ozone generator.
co2-system	Injects carbon dioxide.
air-pump	Air pump.
feeder	Automatic feeder.
sensor	Probe, float switch, leak detector, or other sensor.
power-strip	Switchable power outlets.
algae-scrubber	Grows algae to remove nutrients.
analyzer	Tests water automatically.
ro-di	Reverse osmosis and deionization system, including its valves.
water-changer	Automatic water change system, including its pumps and valves.
alarm-signal	Signals alarms to people or to other systems, such as a siren, a strobe, an indicator light, or a contact to a house alarm panel. <i>Added in 0.3.</i>
fan	Moves air, such as a fan that cools the water by evaporation, or one that ventilates a cabinet or a hood. <i>Added in 0.4.</i>
other	Any other equipment.

Table 25 — Method types

Type	Meaning
sensor	An electronic sensor or meter, attached to the system or handheld.
titration	A titration, by a test kit or an automatic analyzer.
colorimetric-kit	A test kit read by comparing a color by eye.
photometer	A photometer or colorimeter, handheld or in an automatic analyzer.
refractometer	A refractometer.
hydrometer	A hydrometer.

Type	Meaning
icp-oes	Inductively coupled plasma optical emission spectrometry.
icp-ms	Inductively coupled plasma mass spectrometry.
ion-chromatography	Ion chromatography.
calculated	Derived from other values, such as dissolved CO ₂ estimated from pH and alkalinity.
other	Any other method.

Table 26 — Event types

Type	Meaning
water-change	Water was replaced. <code>amount</code> SHOULD give the volume replaced.
dose	A substance was added. <code>amount</code> SHOULD give the volume or mass, and <code>product</code> SHOULD identify the substance. A dose that was due but not given has an <code>amount</code> of 0 (Dosing Section 16.1).
dose-adjusted	The amount a dosing head gives over one dosing day was planned from a test result (Dosing Section 16.2). <code>equipment</code> and <code>channel</code> identify the head. <i>Added in 0.2.</i>
top-off	Water was added to replace evaporation. <code>amount</code> SHOULD give the volume.
refill	A container or reservoir was refilled: <code>equipment</code> and <code>channel</code> for a doser's container, or <code>vessel</code> for a reservoir. <code>amount</code> SHOULD give the volume added, and <code>product</code> what was added. <i>Added in 0.2.</i>
ro-di-run	An RO/DI system made water. <code>time</code> and <code>end</code> give the run, <code>amount</code> the product water made, and <code>wasted</code> the water sent to the drain, where known. <i>Added in 0.2.</i>
feeding	Livestock was fed.
calibration	Equipment was calibrated, or checked against references, such as a probe, a dosing head, or an analyzer. <code>equipment</code> SHOULD identify it, and <code>calibration</code> SHOULD give the record (Section 8.12).
analyzer-test	An analyzer tested the water, or a test that fell due was skipped (Water Testing Section 15.2). <code>equipment</code> identifies the analyzer. <i>Added in 0.2.</i>
maintenance	Equipment was cleaned, serviced, or had media replaced.
livestock-added	Livestock was added. <code>livestock</code> identifies the entry, and <code>count</code> how many were added. <i>Added in 0.2.</i>
livestock-removed	Livestock died or left the system. <code>livestock</code> identifies the entry, <code>count</code> how many, and <code>reason</code> why. <i>Added in 0.2.</i>
alarm	An alarm or a notice was raised. <code>time</code> is when it was raised, and <code>end</code> , when its condition cleared. <code>alarm</code> holds it. <i>Added in 0.2.</i>
power-outage	Mains power was lost. <code>time</code> is when it was lost and <code>end</code> , when known, is when it returned. <code>alarm</code> MAY hold the notice.
restart	Equipment restarted without being asked to, for example after its watchdog. <code>time</code> is when it started again, <code>equipment</code> identifies it, and <code>alarm</code> SHOULD hold the notice. <i>Added in 0.2.</i>
mode	A mode of the automation was active (Automation Section 14). <code>time</code> and <code>end</code> give when, and <code>element</code> identifies the mode. <i>Added in 0.2.</i>
override	Someone overrode an output (Automation Section 15). <code>time</code> and <code>end</code> give when, <code>element</code> identifies the output, and <code>equipment</code> and <code>channel</code> its equipment. <i>Added in 0.2.</i>
program	A step program ran (Automation Section 16). <code>time</code> and <code>end</code> give when, and <code>element</code> identifies the program. <i>Added in 0.2.</i>
note	A free-text note. <code>notes</code> is REQUIRED.

Table 27 — Reference electrodes

Electrode	Meaning
ag-agcl-3m	Silver and silver chloride in 3 mol/L potassium chloride: about +209 mV against the standard hydrogen electrode at 25 °C [USGS].
ag-agcl-3.5m	Silver and silver chloride in 3.5 mol/L potassium chloride: about +205 mV against the standard hydrogen electrode at 25 °C [USGS]. <i>Added in 0.3.</i>
ag-agcl-saturated	Silver and silver chloride in saturated potassium chloride: about +199 mV against the standard hydrogen electrode at 25 °C [USGS].
ag-agcl	Silver and silver chloride, with a filling that is not stated.
she	The standard hydrogen electrode: the value has been converted to it.

Table 28 — Calibration models

Model	Meaning
nernst	An electrode that follows the Nernst equation, such as a pH electrode: <code>offset</code> , in mV, and <code>slope</code> , in % of the theoretical slope (Probe Interface Table 17).
linear	The true value is <code>gain</code> times the reading plus <code>offset</code> : probes other than pH (Probe Interface Table 17), and corrections of an analyzer's results (Water Testing Section 11).
flow-rate	A pump or dosing head that delivers a calibrated flow: <code>rate</code> (Dosing Section 8).

Table 29 — Calibration reasons

Reason	Meaning
points	Too few or too many points.
reference	A reference outside the range the equipment accepts.
spacing	References too close together.
slope	A slope outside its limits.
offset	An offset outside its limits.
gain	A gain outside its limits.
linearity	A point too far from the fitted line.
stability	Readings that did not settle.
timeout	No step was taken for too long, and the calibration was cancelled.
probe	The equipment refused it for a reason of its own, for example because it could not store the result.
person	A person cancelled it.
other	Any other reason.

Table 30 — Livestock kinds

Kind	Meaning
fish	Fish.
coral	Corals and other cnidarians kept as corals, such as anemones and zoanthids.
invertebrate	Other invertebrates, such as shrimps, snails, crabs, and urchins.
plant	Aquatic and marsh plants, including mosses.
macroalgae	Macroalgae, such as those grown in a refugium.
amphibian	Amphibians, such as newts and axolotls.
reptile	Reptiles, such as turtles.
other	Any other organism.

Table 31 — Livestock origins

Origin	Meaning
wild	Collected from the wild.
captive-bred	Bred, propagated, or fragmented in aquariums or hatcheries.
farmed	Grown in ponds, or in the sea from collected stock.

Table 32 — Livestock removal reasons

Reason	Meaning
died	Died, including being euthanized.
rehomed	Given away, sold, traded, or returned.
moved	Moved to another of the owner's systems, where it is recorded as added.
other	Any other reason.

Table 33 — Taxon schemes

Scheme	Meaning
worms	World Register of Marine Species AphiaID, such as 278400 for <i>Amphiprion ocellaris</i> [WoRMS].
gbif	GBIF Backbone Taxonomy key, such as 2353909 for <i>Paracheirodon innesi</i> [GBIF].

Scheme	Meaning
itis	Integrated Taxonomic Information System Taxonomic Serial Number [ITIS].
col	Catalogue of Life identifier [COL].

10 Conformance

10.1 Producers

A conforming producer writes Documents that:

1. are valid against the JSON Schema [JSON Schema] for the declared revision, published in `schema/data-model.schema.json`;
2. meet every requirement of Sections 7, 8, and 9, including those a schema cannot express, such as identifier uniqueness, reference resolution, parameter and unit agreement, and times no more than 60 s later than created; and
3. pass the test procedures of Section 10.4 that apply to producers.

A conforming producer that writes export files writes them as Section 7.8 says.

10.2 Consumers

A conforming consumer:

1. reads every conforming Document of each revision it implements, Documents of earlier revisions and configuration documents as Section 7.7 says, and, if it imports export files, export files as Section 7.8 says;
2. ignores members and extensions it does not understand (Section 7.6);
3. converts units only with the factors in Table 19, and Quantities only with those of Section 7.5;
4. does not treat qualified values as exact (Section 8.4), or the values of a Series with a `statistic` as single readings (Section 8.7);
5. presents each value with its unit;
6. reads a Series as the Observations it is equivalent to (Section 8.7);
7. does not act on a value with an assumed basis until a person has confirmed it (Section 9.4), or on a value whose time is later than its own clock (Section 7.4);
8. treats Documents and export files as untrusted input (Section 11); and
9. passes the test procedures of Section 10.4 that apply to consumers.

10.3 Schema and Validation Tools

The JSON Schema is normative for document structure. Where the schema and this document differ, this document governs. The schema accepts Documents of revisions 0.1 to 0.4, and defines the manifest of an export file as `$defs/exportManifest` (Section 7.8). `registry.json` is normative for the registry, and governs where it and this document differ (Section 9.1).

The validator in `tools/validate_data_model.py` checks Documents against the schema and the additional rules in Sections 7 to 9, and checks that a Document uses nothing that a later revision than the one it declares added. It checks the alarms that events hold against the schema of the Alarms and Notifications specification. It warns about what this Specification advises against, such as a reservoir without `circulating` (Section 8.2), and, in Documents of revisions 0.1 and 0.2, about what revision 0.3 forbids. It checks export files, packed or unpacked: their container, their manifest, and their Document, and their configuration documents and audit log with the validators of their own specifications. Its tests are in `tools/tests/test_validate_data_model.py`. It is informative: a Document is conforming if it meets this Specification, whether or not the tool accepts it.

The validator's checks are part of the [Data Model library for Python](#), which also converts units and Quantities (Sections 7.5 and 9.5), reads Series as Observations (Section 8.7), reads Documents of earlier revisions and configuration documents as Section 7.7 says, reads and writes export files (Section 7.8), merges Documents, and imports and exports CSV. The library is informative as well.

10.4 Test Procedures

Tests of producers export Documents from the product and check them with the validator; tests of consumers import the examples of Appendix B, or copies of them changed as each test says. Each test names the requirements it checks, as sections or as numbered items of sections, and every MUST and MUST NOT of Sections 7 to 11 is checked by at least one test, which `tools/render_spec_tables.py` checks. A pass criterion that checks what a SHOULD asks applies to a product that does it.

Table 34 — Test procedures

Test	Applies to	Requirements	Procedure	Expected result
Valid Documents	Producers	7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7.4, 8.1, 8.2, 8.3, 8.4, 8.5, 8.6, 8.7, 8.8, 8.9, 8.10, 8.11, 8.12, 8.13, 9.2, 9.3, 9.6, 10.1	Export a Document that holds every kind of object and event the product records, in every revision the product writes, and check each with the validator.	The validator reports no errors. No value is recorded both as an Observation and in a Series, and no extension name that begins with <code>org.openaquariumproject</code> is used with a meaning of the product's own.
Earlier revisions	Consumers	7.7.1, 7.7.2, 7.7.3, 8.2.2, 10.2	Import <code>icp-lab-result.json</code> , which declares revision 0.1, and a copy of <code>husbandry.json</code> that declares revision 0.2, without its task by hours of running and the event that completed it, whose new-water reservoir has no <code>circulating</code> , and one of whose events is dated a minute after its <code>created</code> . Show the system's water volume. Then export both as Documents of revision 0.4.	Every object of both is imported, as from Documents of revision 0.4. The reservoir is circulating, and its 60 L count in the system's water volume. The late event is imported. Each export has <code>created</code> , the time it was written, and passes the validator.
Configuration documents	Consumers	7.7.6, 8.2.2	Import, into a consumer that shows a system's water volume, a copy of the Automation example <code>reef-system.json</code> that gives each vessel a volume of 50 L and leaves out its reservoir's <code>circulating</code> . Then import the same copy, declaring Automation revision 0.1.	The first shows a water volume of 100 L, without the reservoir; the second shows 150 L, with it.
Provisional names	Consumers	7.7.5	Import the Alarms and Notifications example <code>history-0.1.json</code> , of revision 0.1.	Its <code>org.openaquariumproject.alarm</code> and <code>org.openaquariumproject.restart</code> events are shown as alarms and restarts, with their alarms.
Extensions	Consumers	7.6.4, 10.2	Import a copy of <code>reef-system.json</code> that gives its system the extension <code>com.example.reef-notes</code> , an object, and a vessel the role <code>com.example.cryptic-tank</code> . Export it again.	The import succeeds and shows what it did without the extension, the vessel is shown with a role the consumer does not know, and the export keeps the extension.
Future times	Producers, consumers	7.4.1, 7.4.2	Enter a manual result dated a day after the producer's clock, and have an analyzer whose clock runs 30 s ahead of the producer's report a result; export at once. Then import a copy of <code>reef-system.json</code> whose last alkalinity result is dated a day after the consumer's clock, and whose <code>created</code> is later still, into a consumer that doses or raises alarms from alkalinity.	The producer does not record the manual result with that date, and records the analyzer's result with its own time; its export passes the validator. The consumer asks a person before it imports the result, and, once it is imported, neither doses nor raises or clears an alarm from it, nor shows it as the latest alkalinity, until its time has passed.
Series	Consumers	8.7.5, 8.7.6, 10.2	Import <code>sensor-history.json</code> , and show the sump temperature of its first Series.	Eleven values, the first at 00:00 and each later one 5 minutes after the one before, with no value at 00:30. The

Test	Applies to	Requirements	Procedure	Expected result
				hourly means of its last Series are shown as means.
State Series	Consumers	8.8.2	Import <code>sensor-history.json</code> , and show how long the heater heated between 00:00 and 01:00.	35 minutes.
Qualified values	Consumers	8.4.3	Import <code>icp-lab-result.json</code> , chart its results, and show their mean together with a later result of the same parameters.	Results below the detection limit are shown with <code><</code> , are not drawn as exact points, and are left out of the mean or make it a limit.
Ranges the owner set	Producers	8.9.6	Set up a new system in the product without setting any range, and export it. Then set an acceptable range for temperature, and export again.	The first export holds no Range; the second holds only the temperature range.
Conversion	Consumers	7.5.2, 8.2, 9.3, 9.5.1, 9.5.2, 9.5.3	Import <code>reef-system.json</code> , and show its alkalinity in meq/L, and the system's water volume in litres. Then import <code>icp-lab-result.json</code> , and ask for its phosphate. Then record a specific gravity of 1.026, and ask for it as salinity.	2.96 meq/L: 8.3 dKH, rounded to its resolution of 0.1 dKH, which is 0.0357 meq/L (Section 9.5, item 3). A water volume of 359.6 L, the 95 gal of the display and the sump, without the top-off reservoir. The total phosphorus <code>element:P</code> is not shown as phosphate, and the specific gravity is not converted to salinity.
Results with no stated basis	Producers	9.4, 9.4.1, 9.4.3	Enter a nitrate result of 10 from a kit whose instructions state results as nitrogen; a nitrate result of 10 from a kit that states no basis, declining to choose one; and a calcium result of 420 from an instrument that labels it "ppm". Export, and check the Document with the validator.	The first is recorded in <code>mg/L-N</code> without <code>unitAssumed</code> ; the second in <code>mg/L-NO3</code> with <code>unitAssumed</code> true, after the producer showed that 10 mg/L as N is 44 mg/L as NO3; the third in <code>mg/L</code> . No unit is "ppm", and the validator reports no errors.
Assumed basis	Consumers	8.4.5, 9.4.4, 9.4.5, 10.2	Import a copy of <code>freshwater-planted.json</code> in which the tap water's nitrate, whose basis was assumed, is moved to the display and dated 19:20, after the display's other nitrate, and which adds an acceptable range for nitrate of up to 20 mg/L-NO3, into a consumer that doses or raises alarms from nitrate results. Then confirm the basis as nitrogen.	The nitrate is shown as having an assumed basis, and no dose, alarm, or correction comes from it. After the confirmation, it is recorded in <code>mg/L-N</code> , with the same value and without <code>unitAssumed</code> , and the consumer acts on it: 10 mg/L as N is 44 mg/L as NO3, above the range.
Totals	Producers	8.6.2	Have a controller record the doses of a head whose dosing day starts at 20:00 as one total for each dosing day, and export a Document of revision 0.4.	Each dosing day is one <code>dose</code> event, whose <code>end</code> is less than 25 hours after its <code>time</code> , and the validator reports no errors.
Running hours	Producers, consumers	8.11.4, 8.11.5	Give a UV sterilizer on a switched output a task with <code>everyHours</code> 9000. Run the output, record the task done, and run it for 8990 hours more, read from its <code>operatingHours</code> , then for 10 more. Export, and import the Document into another consumer that cannot read the output. Then give the task to equipment on an output that reports no <code>operatingHours</code> .	The task is due when the output's <code>operatingHours</code> has risen by 9000 since the event that completed it, and not before. The export holds <code>due</code> , the date the producer expected, and the other consumer shows the task as due on that date. For the last output, the producer counts the time the output reported the equipment running, or tells the owner that it cannot tell when the task falls due.

Test	Applies to	Requirements	Procedure	Expected result
Calibrations	Producers	8.12.5, 8.12.6	Export the calibrations of an analyzer that calibrates itself, which reports no result, and of a pH electrode.	The analyzer's calibration is a completed Calibration Record without <code>result</code> ; the electrode's has its <code>result</code> . No Calibration Record has a <code>time</code> other than its event's, and the validator reports no errors.
Export files	Producers	7.8, 10.1	From a controller, export the history and the configuration as an export file, as a user allowed <code>view-audit-log</code> and <code>export-data</code> , and again as one allowed only <code>export-data</code> . Check each with the validator.	The validator reports no errors. Each file holds one Document, the Automation configuration document, and the alarm settings document, with the same identifiers for the systems and equipment; only the first holds the audit log.
Untrusted export files	Consumers	7.8.8, 11.5	Zip the example export file, <code>examples/export</code> , and import it. Then import copies of it with a member named <code>../evil.json</code> , a member that the manifest does not list, a <code>revision</code> of the Automation configuration that differs from the one the document declares, and a member of 100 MB of zeros, compressed to 100 kB.	The first is imported: its Document as history, and its configuration as configuration. Each copy is refused with a message, no file is written outside the place the consumer unpacks to, and the consumer keeps working.
Untrusted input	Consumers	11.4	Import a Document whose <code>notes</code> hold HTML markup with a script, one whose Series holds ten million values, and one of 100 MB.	The notes are shown as text, and no script runs. The large Documents are refused, or imported up to the consumer's limits, with a message, and the consumer keeps working.

11 Security and Privacy Considerations

Documents can reveal information about their owners. Time zone offsets indicate location; feeding, dosing, and maintenance times, and the states of lights and other equipment, indicate when people are at home; serial numbers identify specific devices; livestock records can reveal valuable animals; and alarms carry the names of the people who acknowledged them.

1. Producers SHOULD let owners review and remove `notes`, `serialNumber`, livestock, State Series, the people named in alarms, and extensions before a Document is shared.
2. Producers SHOULD NOT include personal contact information in Documents.
3. Laboratories SHOULD NOT include customer personal data beyond what the owner needs to match results to a system.
4. Consumers MUST treat Documents as untrusted input: they SHOULD limit document size and the number of values in each Series and State Series, MUST NOT execute any content, and MUST escape strings before displaying them.
5. Consumers MUST treat export files as untrusted input too: they MUST NOT write a member outside the place they unpack the file to, whatever its name, and SHOULD limit how much they unpack, because a small ZIP file can expand to fill a disk. A configuration in an export file changes what equipment does, so a consumer SHOULD show it to a person before it applies it.

Documents do not carry the audit log of the Security and Access specification [OAP-SA Section 12]. The audit log has rules that Documents cannot keep: its records are never changed, are deleted only when the oldest are dropped or the unit is reset to factory settings, are viewed only by users allowed the `view-audit-log` operation, and name credentials and roles. Documents are edited, merged, shared, and erased by their owners. A product that exports history SHOULD offer to put the audit log beside it in an export file, in that specification's format, for a person who may view it (Section 7.8, item 7).

This revision does not define a way to sign Documents or export files, or to verify who produced them.

Appendix A — Approval Checklist

Complete this checklist before requesting Working Group approval. Link each item to the evidence.

Item	Status	Details
Section 2 describes compliance with all five tenets.	Done	
A public review period of at least 30 days has been announced and completed.	Not started	
At least two independent implementations exist, at least one from an organization not represented among the Authors.	Not started	
The implementations have exchanged Documents successfully.	Not started	
Every normative reference is publicly available.	Done	
The schema, registry, and examples are published, and the examples pass the validator.	Done	
Test procedures are published.	Done	Section 10.4.
Appendix C is empty, or each open issue is explicitly deferred.	In progress	Revision 0.4 resolves every open issue but issue 24, which waits on the documentation of test kits; issues 7, 8, 27, 29, 30, and 36 are deferred beyond revision 1.0.
No unresolved Exclusion Notices for this Specification are recorded in <code>Notices.md</code> .	Done	
The Maintainer has recorded consensus to approve.	Not started	

Appendix B — Examples (Informative)

A Document with one system and one observation:

```
{
  "oapDataModel": "0.4",
  "created": "2026-10-13T08:00:00+10:00",
  "systems": [
    {
      "id": "pond",
      "name": "Garden pond",
      "water": "freshwater",
      "tags": ["pond"],
      "vessels": [{ "id": "pond-main", "role": "display" }]
    }
  ],
  "observations": [
    {
      "parameter": "temperature",
      "value": 14.5,
      "unit": "degC",
      "time": "2026-10-13T07:30:00+10:00",
      "system": "pond"
    }
  ]
}
```

The `examples` directory contains complete Documents:

- `reef-system.json`: a reef system with a sump and top-off reservoir, controller sensor readings, test kit results, a water change, a dose from one doser channel, and a pH calibration.
- `sensor-history.json`: a controller's sensor history as Series, at a fixed interval, at irregular times, and as hourly means; and the states of a heater, a return pump, a top-off float switch, and a leak sensor as State Series, with the feed mode that changed the pump.
- `controller-history.json`: a controller's events: alarms, a power outage, and a restart, with their alarms; a day's top-off fills as one total; a doser's doses of a dosing day that crosses midnight as one total; a reservoir refill; an RO/DI run with its waste water; a doser's adjusted amount and container refill; an analyzer test; and a mode, an override, and a step program; with a siren of the category `alarm-signal`.
- `calibrations.json`: calibration records of a pH electrode, completed and failed, a conductivity cell, a dosing head, an ORP check, an analyzer's comparison with a laboratory result and its correction, and the analyzer's own calibration, which reports no result; a corrected result and a reference result; a pH electrode's end of life; and specific gravity with its reference temperature.

- `husbandry.json`: target ranges, livestock with taxonomic identifiers and the events that added and removed them, maintenance tasks by date and by hours of running and the events that completed them, an RO/DI source vessel, a new-water reservoir that is not circulating, and products with GTINs.
- `icp-lab-result.json`: a laboratory ICP result, with results below the detection limit. It declares revision 0.1, to show that Documents of revision 0.1 remain valid; the Dosing specification's `history.json` declares revision 0.2.
- `freshwater-planted.json`: a planted freshwater system using °F, dKH, dGH, and UUID identifiers, with fish, shrimp, and plants, and tap water whose nitrate has an assumed basis.
- `minimal.json`: the Document above.
- `export`: an export file, unpacked: its manifest; a controller's history; its Automation configuration document and alarm settings document, copies of those specifications' reef examples, whose systems and equipment have the same identifiers; and an audit log, a copy of the Security and Access example. Packed as a ZIP file, with `manifest.json` at its root, it is a complete export file.

Appendix C — Open Issues (Informative)

Issues 1 to 23 were open in revision 0.1, and issues 24 to 34 arose in revision 0.2. The Project Lead decided issues 33 and 34 on 15 October 2026, and revision 0.3 applies them. Issues 35 to 39 arose in revision 0.3. On 17 October 2026 the Project Lead froze the scope of revision 1.0, and decided issues 6, 10, 25, 28, 31, 37, and 38; revision 0.4 applies the decisions, and resolves issues 32 and 39, which other specifications raised. Issue 40 records a choice that revision 0.4 made where those decisions left it open, which the Project Lead accepted on 18 October 2026. Issues 7, 8, 27, 29, 30, and 36 are deferred beyond revision 1.0, and are tracked outside this Specification. Issue 24 stays open: it waits on the documentation of the test kits that owners use. Each issue says how a revision resolves it, or why it stays open. Issue numbers do not change, because other specifications refer to them.

1. **Compact time series.** *Resolved in revision 0.2.* Series (Section 8.7) hold values at a fixed interval or at irregular times, and hourly or daily means, minima, and maxima.
2. **Livestock.** *Resolved in revision 0.2.* Livestock entries (Section 8.10) identify taxa by scientific name and by identifiers in WoRMS, GBIF, ITIS, or the Catalogue of Life, and events record livestock added and removed.
3. **Specific gravity reference temperature.** *Resolved in revision 0.2.* A Method's `referenceTemperature` records it (Section 8.4).
4. **Unstated chemical basis.** *Resolved in revision 0.2, with provisional values.* Section 9.4 says what producers do, and `unitAssumed` marks an assumed basis. Revision 0.3 forbids acting on such a value until a person has confirmed its basis, and says how a confirmation is recorded. The assumed units of Table 21 are provisional (issue 24).
5. **Lighting.** *Resolved in revision 0.2.* The `Lighting Programs` specification answers it (its Section 19): the Data Model records light as measured, as `ppfd` and `daily-light-integral` observations; programs are configuration, which the Automation specification's documents carry; and spectra belong to the light's maker.
6. **Schema identifier.** *Resolved in revision 0.4, as the Project Lead decided on 17 October 2026.* The schema's `$id` was the placeholder `urn:x-oap:schema:data-model:0.3`, until the project had a permanent web address. Revision 0.4 gives it `https://openaquariumproject.org/schemas/data-model/0.4`, the form that every specification's schemas take, and the project's website publishes each schema at its `$id`.
7. **Source water.** *Resolved in revision 0.2.* Stored water, such as new saltwater, remains a `reservoir`. Water the system does not store, such as tap water or RO/DI product water, is a vessel with the role `source` (Section 8.2).
8. **Requirement identifiers.** *Resolved in revision 0.4, as the Project Lead decided on 17 October 2026: no separate identifiers.* Numbered requirements would help conformance test suites, but they should follow one convention across the project's specifications. Tests name the requirements they check by section and numbered item (Table 34), as every specification's tests do, and that is the convention.

9. **State history.** *Resolved in revision 0.2.* State Series (Section 8.8) record the states of outputs and inputs over time.
10. **Target ranges.** *Resolved in revision 0.2.* Ranges (Section 8.9) are the ranges that the [Alarms and Notifications](#) specification proposed (its Sections 9.6 and 20.3), without its [delay](#) and [hysteresis](#).
11. **Configuration.** *Resolved in revision 0.2.* Configuration moves in the [Automation](#) specification's configuration documents and the Alarms and Notifications specification's alarm settings documents, which use the identifiers of a Document's systems and equipment, rather than in Documents. History records modes, overrides, and step programs as events that name their elements (Section 8.6). A single file that holds a Document and its configuration documents together is issue 25.
12. **Reminders.** *Resolved in revision 0.2.* Tasks (Section 8.11) record maintenance, when it falls due, and when to remind people; events complete them. How reminders are delivered is issue 26.
13. **Calibration records.** *Resolved in revision 0.2.* Calibration Records (Section 8.12) hold points, references and their temperatures and solutions, readings, results as offset and slope, gain, or flow rate, the result replaced, the outcome, and the reason. A probe's age comes from its Equipment's [installed](#), and its expected end from [endOfLife](#).
14. **Audit records.** *Resolved in revision 0.2.* Documents do not carry audit records, which keep their own format and rules; a product offers to export the audit log beside history (Section 11). Raised by the [Security and Access](#) specification (Appendix D, issue 11).
15. **Alarm history.** *Resolved in revision 0.2.* The event types [alarm](#) and [restart](#), and the Event member [alarm](#) (Section 8.13), as the Alarms and Notifications specification proposed (its Section 20.3). The provisional names it used are read as these (Table 3).
16. **Water level records.** *Resolved in revision 0.2.* Top-off is recorded one fill at a time, or as one event a day that totals the fills, with [occurrences](#) (Section 8.6). A refilled reservoir is a [refill](#) event with its [vessel](#). An RO/DI run is an [ro-di-run](#) event, with the product water as [amount](#) and the water sent to the drain as [wasted](#); flushes during a run count in its [wasted](#), and a flush on its own is a [maintenance](#) event. New water measured before a water change remains an observation of its reservoir, and the water an RO/DI unit makes can be an observation of a [source](#) vessel (issue 9). Raised by the [Water Level and Water Change](#) specification (Appendix D, issue 11).
17. **Dosing records.** *Resolved in revision 0.2.* The event types [dose-adjusted](#) and [refill](#) replace the provisional ones (Table 3). Doses are recorded one by one, or as one event a day that totals a head's given doses, with [occurrences](#); missed and skipped doses are always recorded one by one. Calibrations of heads fit Calibration Records with the [flow-rate](#) model. The details of doses and plans stay in the Dosing specification's extension (Table 11). Products carry GTINs; a product registry remains open (issue 8). Raised by the [Dosing](#) specification (Appendix D, issue 10).
18. **Water testing records.** *Resolved in revision 0.2.* The event type [analyzer-test](#) records tests; an observation's [correction](#) holds its raw value and the correction applied, and [reference](#) marks results used as references; comparisons are Calibration Records with the outcome [checked](#), and corrections Calibration Records with the [linear](#) model. A suspect result that a later one confirms stays in the Water Testing specification's own record, because only its screening produces such results. Raised by the [Water Testing](#) specification (Appendix D, issue 10), whose revision 0.2 uses them.
19. **Lighting records.** *Resolved in revision 0.2.* The parameter [daily-light-integral](#), in [mol/m2/d](#). Raised by the [Lighting Programs](#) specification (Appendix D, issue 5).
20. **Flow records.** *Resolved in revision 0.2.* The parameter [turnover](#), in [1/h](#), and a Vessel's [dimensions](#) (Section 8.2). Raised by the [Flow Programs](#) specification (Appendix D, issue 6).
21. **Probe calibration records.** *Resolved in revision 0.2.* Calibration Records hold a probe calibration's points, result, the result it replaced, outcome, and reason; the probe type, compensation method, nominal cell constant, factor for total dissolved solids, and shift stay in the Probe Interface specification's extension (Table 11). A Method's [referenceElectrode](#) records the reference of an ORP reading, its [referenceTemperature](#) the reference temperature of specific gravity, and an Equipment's [endOfLife](#) a predicted end of life. Raised by the [Probe Interface](#) specification (Appendix D, issue 8), whose revision 0.2 uses them, with the reference electrode [ag-agcL-3.5m](#) that revision 0.3 adds.
22. **Assumed units.** *Open: waits on the documentation of the test kits that owners use.* The units that Table 21 assumes for results with no stated basis need checking against the documentation of the test kits

owners use, especially for ammonia, where kits that name both NH_3 and NH_4^+ may report as either, or as nitrogen. For revision 0.4, the published instructions, color charts, and pages of about 95 test kits, checkers, and test strips from 13 makers were read. Every one that reports as nitrogen or phosphorus says so; the kits that state a basis report nitrate, nitrite, and phosphate as the ion, silicate as SiO_2 , and alkalinity and hardness in “ppm” as calcium carbonate, as Table 21 assumes; and most kits print “ppm” or “mg/L” beside the ion’s name without saying what the number is expressed as. Of the ammonia kits that name both NH_3 and NH_4^+ and show a basis, half report as NH_3 and half label their results NH_4 , so the assumed unit is right, or 6 % off, for each of them. Still to be checked are the printed color charts of several kits, the booklets of one maker’s multi-test kits, leaflets that only retailers publish, the species behind one maker’s silicate results, which its conversion factor suggests is SiO_4 , and the basis of the one free-ammonia indicator found.

23. **One file for an export.** *Resolved in revision 0.4, as the Project Lead decided on 17 October 2026.* A complete export is a Document with the configuration documents of the Automation and Alarms and Notifications specifications, and perhaps the audit log. Revision 0.4 defines an export file: a ZIP file with a manifest that holds them together, so that moving to a new product is one step (Section 7.8).
24. **Reminder delivery.** *Resolved: the Alarms and Notifications specification adds the notice `task-due`, as the Project Lead decided on 17 October 2026.* Tasks have due dates and reminder leads, but no alarm code. The Alarms and Notifications specification adds a `task-due` notice, `info` by default, raised at a task’s reminder lead, as the Water Testing specification added `test-due` (Section 8.11, item 7).
25. **Equipment categories.** *Resolved in revision 0.4, as the Project Lead decided on 17 October 2026.* The Alarms and Notifications specification needed an `alarm-signal` category (its Appendix D, issue 4), and the Automation specification a `fan` category (its Appendix D, issue 4). Each needed a Fail-Safe Behavior recommendation, which is a Project Lead decision. The Project Lead decided the recommendation for `alarm-signal` (Fail-Safe Behavior Appendix C, issue 10), and revision 0.3 adds it. On 17 October 2026 the Project Lead decided the recommendation for `fan`, which Fail-Safe Behavior revision 0.4 gives, and revision 0.4 adds the category (Table 24).
26. **Significant digits of converted values.** *Resolved in revision 0.4, as the Project Lead decided on 17 October 2026.* Section 9.5, item 3, gave a converted value no more significant digits than the original had, so 8.3 dKH was shown as 3.0 meq/L, while the example of Section 9.5 computes 2.96 meq/L. Counting significant digits also loses resolution when a conversion lowers the leading digit: 3.0 meq/L is 8.4 dKH. Revision 0.4 rounds a converted value to the original’s resolution, converted: 0.1 dKH is 0.036 meq/L, which gives 2.96 meq/L (Section 9.5, item 3). The Conversion test of Table 34 and the [Data Model library](#) follow.
27. **Converting quantities.** *Resolved in revision 0.4.* The registry gave factors for the units of parameters, but none for the volume, mass, and length units of Quantities. A consumer that adds up the volumes of a system’s circulating vessels (Section 8.2), one in `gal` and another in `L`, had no factor to convert with, although Table 20 gave the sizes of `gal`, `gal-imp`, and `in` in words. Revision 0.4 gives each of these units a factor to the base unit of its kind, `L`, `g`, or `m` (Section 7.5).

Decided by the Project Lead

33. **Which source governs.** *Resolved in revision 0.3.* The Project Lead decided on 15 October 2026 that `registry.json` governs where it and this document differ, as the Device Classes specification says of its `classes.json`. Section 9.1 says so, and Section 10.3 makes `registry.json` normative.
34. **Reservoirs that do not circulate.** *Resolved in revision 0.3.* The Project Lead decided on 15 October 2026 that, from revision 0.3, `circulating` defaults to `false` for `reservoir`, as it does for `source`, so that a reservoir left without it does not inflate a system’s volume, and the range alarms and dosing computed from it. Documents of revisions 0.1 and 0.2 keep the old default, and the validator and library warn when a reservoir leaves it out (Sections 7.7 and 8.2). The library’s `upgrade` gives such reservoirs `circulating true`.

The following issues arose in revision 0.3:

35. **Results that are not to be acted on.** *Resolved: the Automation specification’s revision 0.3 makes the change.* Section 9.4 forbids acting on a value with an assumed basis until a person confirms it, and Section 7.4 on a value dated later than the consumer’s clock. The Automation specification’s `observation`

input [OAP-AU Section 9.5] read the latest observation of a parameter; its revision 0.3 passes over such observations. The Water Testing specification raised the same question (its Appendix D, issue 32).

36. **Clocks that differ.** *Resolved in revision 0.4, as the Project Lead decided on 17 October 2026.* Section 7.4 refused any time later than `created`, but the clocks of the instruments that took the values may run a few seconds ahead of the producer's. Revision 0.4 accepts times up to 60 s later than `created`, the bound that the Fail-Safe Behavior specification puts on the error of a valid device clock [OAP-FS Section 10.5.4]. It also makes `created` REQUIRED, so that a validator can check every time against it.
37. **Calibrations that report no result.** *Resolved in revision 0.4, as the Project Lead decided on 17 October 2026.* A Calibration Record with the outcome `completed` needed its `result` (Section 8.12), but an analyzer does not report what its own calibration computed, so the Water Testing specification recorded such a calibration only in its own extension (its Appendix D, issue 31). Revision 0.4 lets a completed record leave out `result` for equipment that does not report one (Section 8.12, item 5).
38. **The time of a Calibration Record.** *Resolved in revision 0.4.* A Calibration Record takes its time from its event (Section 8.12), but the Device Classes attribute `lastCalibration` and a smart probe's record hold Calibration Records without an event, so the Probe Interface specification gave each record its time in its own extension (its Appendix D, issue 29). Revision 0.4 adds an OPTIONAL `time`, when the calibration ended, for records outside events.

The following issue records a choice of revision 0.4:

40. **Choices of revision 0.4.** *Resolved in revision 0.4: the Project Lead accepted the choice on 18 October 2026.* Section 7.4 compared times with `created`, which was OPTIONAL, so a Document without it could hold any future time, and neither the validator nor the Valid Documents test could catch it. Revision 0.4 makes `created` REQUIRED (Table 1), rather than comparing times with when a Document is written, which no tool can check; Documents of earlier revisions stay valid without it (Section 7.7).