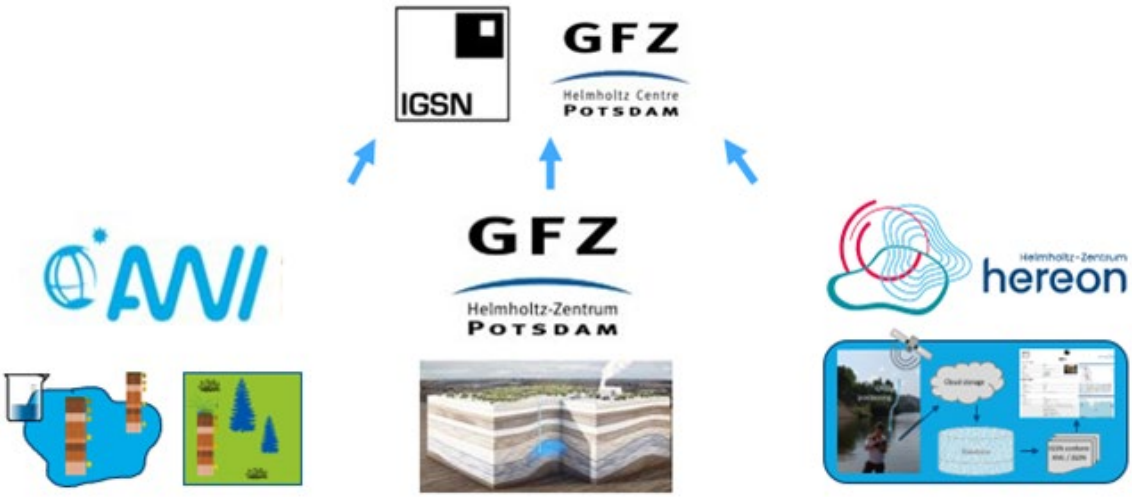
FAIR Workflows to establish IGSN for Samples in the Helmholtz Association (FAIR WISH)

D8 - Scripts for generation of metadata in XML and JSON – documentation



Authors

Simone Frenzel (GFZ)

Kirsten Elger (GFZ)

Citation of this document:

Frenzel, S., Elger, K. (2023). FAIR WISH D8 - Scripts for generation of metadata in XML: documentation. Zenodo. https://doi.org/10.5281/zenodo.10443226

Citation of the software described in this document:

Frenzel, S. (2023): FAIR WISH Software Tool: SAMIRA: FAIR SAMPLES Template Processing. V. 1.0. GFZ Data Services. https://doi.org/10.5880/GFZ.LIS.2023.001

Table of contents

1. Introduction

1.1. Purpose of this document

This document describes the deliverable “D8 - Scripts for generation of metadata in XML and JSON (GFZ)” of the project FAIR Workflows to establish IGSN for Samples in the Helmholtz Association (FAIR WISH) funded by the Helmholtz Metadata Collaboration (HMC).

This deliverable is partly based on the FAIR WISH: Sample description template (Brauser et al., 2023). The SAMIRA software, described here, is published with the DOI http://doi.org/10.5880/GFZ.LIS.2023.001 (Frenzel 2023).

1.2. Software for assigning PIDs for physical samples/ specimen

Physical samples (or specimen or artefacts) represent the origin of research results in many scientific disciplines. Assigning persistent identifier (PID) to samples is a fundamental step to make them discoverable and traceable in unambiguous way over the Web. The International Generic Sample Number (IGSN) is a PID for physical samples and connecting these with their online description following a dedicated metadata schema.

Sample descriptions of samples are available in various formats and detail. In order to publish them in a standardized manner and to automate and standardize the preparation and processing, the software product SAMIRA (Sample IGSN Registration Automation) was created as part of the Project FAIR WISH, funded by the Helmholtz Metadata Collaboration (HMC).

SAMIRA aims to automate the generation of Metadata XML-Files for the Registration of PID from different input sources (e.g. the FAIR Samples Template, Wiezcorek et al., 2023). This first version of SAMIRA implements the creation of IGSN and Datacite metadata and the respective PID registration.

2. Sample description metadata schemes

The metadata schemas describing physical samples, as defined by the IGSN e.V. are based on the following XML schemes.

igsn.xsd

Describes registration information, for example the registration date.

resource.xsd

Describes the basic information of each IGSN such as the 'sampleAccess' and serves as a kernel or so-called birth certificate, which with 'supplementalMetadata' provides the opportunity to describe the sample in a modular manner.

sample.xsd

Is one of the modular versions to describe samples in the earth and environmental context

The IGSN DataCite Partnership (Buys and Lehnert, 2021) resulted in a change in the persistent identifier (PID) type: All <10 Mio IGSNs that were originally registered as Handle PIDs had to be re-registered as DataCite DOIs (IGSN IDs) and their metadata was partly mapped to the DataCite Schema (metadata.xsd) following the agreed recommendations between the IGSN DataCite Partnership Steering Group (https://support.datacite.org/docs/igsn-id-Metadata-recommendations).

Extensions and changes to schemas are crucial to this software development process. SAMIRA's implementation is generic and allows any schema to be represented as an object.

3. Selection of technical tools

The following criteria were crucial for selecting the technical tools for implementation. Existing implementations, performance, future team competence, maintainability.

Languages	**Java**	**PHP**	**Python**
Existing implementations	Outdated, no longer maintained customized DataCite metadata store	Used for the majority of GFZ Data Services codes. Connectable with sample management software (e.g. mDIS, Behrends et al., 2020)s	Nothing known
Performance	very high	Mediocre; Improved with php 8.0	Low
Future team competence	Existing: probably difficult in the future, needs specialized software developer team	Existing: is part of the Potsdam University of Applied Sciences curriculum for future specialist staff	Existing, not further considered due to previous factors
Maintainability	Needs specialized software developer team	Can be easily learned and errors can be easily fixed at runtime	Not considered further

Java has good XML support and PHP predominates in the other aspects. Below is a closer look at the PHP extension for XML generation tor the decision.

PHP XML Extension Selection

PHP XML Extension [1]	**XMLWriter**	**SimpleXML**	**DOM**
Performance	Very good	Good - mediocre	Slow
Maintainability - Readability	Depends on tidy and carefully structured code	Easy	Not considered further due to previous factors
Maintainability - Further development adaptability	easy	Difficult with prescribed arrangement of XML elements, may require complete re-implementation	Not considered further due to previous factors

[1] https://p0l0.binware.org/2011/07/04/simplexml-vs-xmlwriter-vs-dom/

In the long term, a PHP application that combines the advantages of XMLWriter and SimpleXML is the most reasonable implementation. Based on this decision the following core classes identified (Figure 1)

4. Implementation

4.1. Classes

Figure 1: SAMIRA core classes

The classes Element.php and IGSN-Metadata_Base_Representation.php are created for an abstract description of metadata information.

Element class

Describes metadata information with the properties: $keyname; $value; $mandatory, $hasattributes, $isdependent, $isparentstag, $children, $allowedvaluelist, $attributes to enable all possibilities for representing metadata information in an XML node or XML tag.

IGSN-Metadata_Base_Representation class

Superclass that provides the methods for setting elements in all inheriting metadata classes: For example: public function setbasicElement($value, $keyname) which creates a simple XML tag with the desired name and value.

IGSN-Metadata_Base_XMLWriter class

Superclass that provides the methods for writing the properties of the metadata representation classes as an XML string by calling the required functions of the PHP XMLWriter module in a loop

Inheriting metadata representation classes

Each necessary schema is implemented as a class which represents the respective metadata descriptions as a property of type of the Element.php class.

Inheriting metadata XMLWriter classes:

Currently due to the existence of the various namespace specifications and additionally required information.

4.2. SAMIRA and the FAIR SAMPLES Template

When registering DOIs, both via the IGSN Metadata Store and the DataCite rest API, the metadata is first transmitted and then the URL of the landing page. This can result in the metadata of an already registered IGSN/ DOI to be overwritten. SAMIRA includes functions that check for duplicates before registration.

Incorrect information

Incorrect information is communicated to the user and recorded in a log file after the XML files are validated against the schema: The program stops here because incorrect information needs to be curated by a human. SAMIRA includes classes for validation and error logging.

Input errors

Input errors such as incorrect capitalization and unnecessary spaces/ blanks are fixed in SAMIRA during the csv input step..

Multiple entries in cell

Sometimes, specific metadata properties, like “related identifiers” or “contributors” can occur more than one time. In the FAIR SAMPLES Template, this information should be provided in one field as a list with agreed delimiter. Subclasses of these properties, like “last name”, “given name”, ORCID or “affiliation” for properties have to be provided similarly in the respective cells (for all contributors separated by the agreed delimiter).

SAMIRA divides the information using agreed delimiter and uses functions in helper scripts to create arrays that are used to create XML nodes with parent and child elements including the required attributes (properties and sub-properties)

4.3. IGSN Registration

For registration in the IGSN Metadata Store and Datacite, all functions for communication with the corresponding REST interfaces have been implemented in the Classes IGSNRestRequest.php and DataCiteRestRequest.php. These functions enable mining an IGSN and updating the various metadata XML files individually or in combination.

5. Changes to the original project proposal

The partnership between IGSN and DataCite made large changes to the overall IGSN system, which had a decisive impact on the design of this diversified system and its further development. Among other things, the implementation of a JSON variant was omitted at this stage, because the creation of a JSON file from the associative array created during the reading of the csv is possible, but proved to be of little use without a corresponding schema definition.

6. Outlook

SAMIRA is intended to be used in various application scenarios, specifically as a stand-alone program, as a Php module and as a web API, which requires different versions that are implemented via parameterization or configuration information.

The further enrichment of the the Datacite metadata schema for IGSN-related sample descriptions is continuously (beyond the mandatory fields) is continually being continued based on the DataCite Recommendations for IGSNs.

Schema extensions and new schemas that become necessary as part of the IGSN registration will be maintained in SAMIRA.

7. References

Frenzel, Simone (2023): FAIR WISH Software Tool: SAMIRA: FAIR SAMPLES Template Processing. V. 1.0. GFZ Data Services. https://doi.org/10.5880/GFZ.LIS.2023.001

P0L0’a Blog: SimpleXML vs XMLWriter vs DOM. URL: https://p0l0.binware.org/2011/07/04/simplexml-vs-xmlwriter-vs-dom/ (last access 29 Dec 2023)

Wieczorek, M., Brauser, A., Frenzel, S., Heim, B., Baldewein, L., Kleeberg, U., & Elger, K. (2023). FAIR WISH “FAIR SAMPLES Template”. Zenodo. https://doi.org/10.5281/ZENODO.7520015