3rd FAIR Digital Objects Conference

Lightning talk

TIB-OP_will_set_DOI

Published: 1970-01-01

Typed PID Maker

Enabling Management of Typed and Validated FDO Records

Andreas Pfeil[1], Maximilian Inckmann[1], and Thomas Jejkal[1]

[1]Karlsruhe Institute of Technology, Germany

*Correspondence: Andreas Pfeil, andreas.pfeil@kit.edu

Abstract. FDO records contain the most high-level set of metadata about an object in the concept of FDOs. Each attribute in a record has a definition, conveying semantical and structural information, and a value. Therefore, the allowed values for an attribute are well-defined, and the definition can be used for validating a value. It can be considered a good practice validating record information before publishing it. We introduce the Typed PID Maker, which exposes a REST-API for FDO record management, where validation will be performed on each action that aims to create or modify a record.

Keywords: FAIR Digital Objects, Validation, Handle System, Persistent Identifiers

1. Extended abstract

From the beginning, FAIR Digital Objects (FDOs) are designed as machine-actionable entities. Machine-actionability is heavily dependent on a strong typing mechanism for attribute values in FDO records, combined with semantics [1]. Similar to compilers of programming languages, which perform type-checking before the actual compilation, record information should be validated before being published in FDO records.

Validation requires an FDO record that comes with certain information. In order to bind the lifetime of an FDO record to the lifetime of its PID, a straightforward approach is to represent FDO records within the PID system. The Handle system is a highly reliable and well-established system that offers globally unique, resolvable, and persistent identifiers (PIDs) [2], as needed for FDOs. In fact, it is the technological basis for the DOI system [3]. The Handle system also provides a record structure for each PID, that can be described as a list of tuples. This matches the requirements for FDO records very well [4]. The first element of a tuple refers to an attribute definition using a PID. These definitions are FDOs that exist in a registry. The second element of a tuple contains a value that must respect the restrictions imposed by its respective attribute definition. With this information, each attribute can be validated independently. Validation will ensure both, that a proper type definition is available, and that the value is valid according to the definition.

So-called Kernel Information Profiles specify required and optional attributes, and FDO records can claim compliance with a profile. These claims can and should be validated before creating or updating an FDO record. To make such a claim, a record

must contain a tuple, consisting of a reference to a well-known attribute definition [5] and a PID that references the profile. This tuple is all information required to make profile validation possible. One profile is the Helmholtz Kernel Information Profile [6], which contains basic attributes and is based on recommendations from the Research Data Alliance [7]. First registries for attribute definitions and profiles are available [8], [9].

Validation is a complex process that involves read access to the Handle system and registries. In order to ensure that validation is always performed before creating or updating a PID record, we created the Typed PID Maker. It provides a REST-API that enables validated write access to the Handle system and PID resolution with optional validation. For write access, it must be configured to a Handle prefix, including authentication information. The Typed PID Maker also supports other PID-related tasks, such as PID logging and record search, PID suffix generation, and a sandbox mode for test environments. The sandbox mode enables users to explore the functionalities of Typed PID Maker and its validation mechanism without creating actual Handle PIDs. It can also be used to test the development of applications that create or process PID records in a safe environment.

An instance of the Typed PID Maker in sandbox mode is publicly available and free to explore [10]. It runs the latest development preview, which adds batch operations, improved performance and validation and support for the Type API [11]. We aim to continue evolving and aligning the Typed PID Maker with the developments in the field and upcoming best practices.

Data availability statement

This work is not based on specific data. It introduces a software for FDO record validation, management, and related tasks. As this work represents a software development project, its data consists of publicly available source code on GitHub [12]. The source code of every release is published on Zenodo[13].

Underlying and related material

A preview instance of the Typed PID Maker is available. It is in sandbox more, not creating real PIDs and removing all its data every 24 hours at 03:30 CET [10].

Author contributions

Thomas Jejkal conceptualized and developed an early version of the Typed PID Maker, and is responsible for the overall supervision and guidance of the project. Andreas Pfeil overtook the development in an early state and is since the main developer. Maximilian Inckmann contributed new features to the newest development version. All authors discussed the results and contributed to the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Funding

This research has been supported by the research program 'Engineering Digital Futures' of the Helmholtz Association of German Research Centers and the Helmholtz Metadata Collaboration Platform (HMC). It is also supported by the NFDI-MatWerk consortium, funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under the National Research Data Infrastructure – NFDI 38/1 – project number 460247524.

Acknowledgements

The authors thank all contributors from the Research Data Alliance, the FAIR DO Forum, and HMC, as well as potential users, for fruitful discussions and feedback. We especially thank the authors of the report of the working group on Persistent Identifier Information Types of the Research Data Alliance[14], on which the initial implementation of the Typed PID Maker was based on.

References

References

[1] W. Claus, I. Sharif, B. Daan, A. Ivonne, and W. Peter, "FDO machine actionability," Nov. 19, 2022. DOI: 10.5281/zenodo.7825650. Accessed: Nov. 2, 2025. [Online]. Available: https://zenodo.org/records/7825650.

[2] S. Sun, L. Lannom, and B. P. Boesch, "Handle system overview," Internet Engineering Task Force, Informational RFC 3650, Nov. 2003. DOI: 10.17487/rfc3650. [Online]. Available: https://datatracker.ietf.org/doc/rfc3650/.

[3] *DOI Handbook*. DOI: https://doi.org/10.1000/182. Accessed: Oct. 31, 2025.

[4] A. Ivonne et al., "FDO PID profiles & attributes," Oct. 17, 2022. DOI: 10.5281/zenodo.7825630. Accessed: Nov. 2, 2025. [Online]. Available: https://zenodo.org/records/7825630.

[5] "KernelInformationProfile attribute definition," Accessed: Nov. 2, 2025. [Online]. Available: https://dtr-test.pidconsortium.net/#objects/21.T11148/076759916209e5d62bd5.

[6] C. Curdt et al., "Helmholtz Metadata Collaboration, Helmholtz Kernel Information Profile," en, HMC Office, GEOMAR Helmholtz Centre for Ocean Research, Kiel, Germany, Report, Dec. 2022. DOI: 10.3289/HMC_publ_03. Accessed: Jul. 15, 2024. [Online]. Available: https://oceanrep.geomar.de/id/eprint/57942/.

[7] T. Weigel et al., "RDA Recommendation on PID Kernel Information," en-US, Research Data Alliance, Recommendation, Nov. 2019. DOI: 10.15497/rda00031. Accessed: Mar. 27, 2025. [Online]. Available: https://zenodo.org/records/3581275.

[8] "Data Type Registry test instance," Accessed: Nov. 2, 2025. [Online]. Available: https://dtr-test.pidconsortium.net/#urls/intro.html.

[9] "Typeregistry lab," Accessed: Nov. 2, 2025. [Online]. Available: https://typeregistry.lab.pidconsortium.net/.

[10] "Typed PID Maker preview instance – API documentation," Accessed: Nov. 2, 2025. [Online]. Available: https://typed-pid-maker.datamanager.kit.edu/preview/swagger-ui/index.html.

[11] H. Lienhop and FC4E-T4-3, *TypeAPI*, Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen, Göttingen, Germany, Apr. 2025. Accessed: Oct. 31, 2025. [Online]. Available: https://github.com/FC4E-T4-3/type-api.

[12] A. Pfeil and T. Jejkal, *Typed PID Maker*, Dec. 2022. Accessed: Oct. 31, 2025. [Online]. Available: https://github.com/kit-data-manager/pit-service.

[13] A. Pfeil and T. Jejkal, *Typed PID maker*, Jun. 12, 2025. DOI: 10.5281/zenodo.15650488. Accessed: Nov. 2, 2025. [Online]. Available: https://zenodo.org/records/15650488.

[14] T. Weigel, T. DiLauro, and T. Zastrow, "PID information types WG final deliverable," Oct. 10, 2015, Publisher: Zenodo. DOI: 10.15497/FDAA09D5-5ED0-403D-B97A-2675E1EBE786. Accessed: Nov. 2, 2025. [Online]. Available: https://zenodo.org/records/1405977.