

# Building and Maintaining a Registry for PID Info Types

A DTR of the ePIC Persistent Identifier Consortium

Ulrich Schwardmann, GWDG

Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen  
(GWDG)

Am Fassberg, 37077 Göttingen  
ulrich.schwardmann [at] gwdg.de

17 September 2016, Denver

# ePIC PIT-DTR registration GUI

see: <http://dtr.pidconsortium.eu:8081>  
currently 75 PID-BasicInfoTypes and 57 PID-InfoTypes defined



Ulrich  
Schwardmann,  
GWDG

DTRs and  
PID Info  
Types

Type  
Hierarchies

Schemas

Type  
Validation

Discussion

A screenshot of a web browser window displaying the Document Repository interface. The browser's address bar shows the URL "dtr.pidconsortium.eu:8081/#objects/?query=time". The page title is "Document Repository" and the search bar contains the text "time". Below the search bar, there are navigation links for "Prev", "1", "2", and "Next", and a "JSON View" button. The main content area displays a list of search results for the query "time". The results are listed as follows:

- full-time** [Type: PID-BasicInfoType]  
Description: time representation as string (RFC3339; ISO 8601), defined by regular expression, because JSON "time"-format allows "11:11:11Z" or "11:11:11+11:11". "11:11:11Z" is seen as "11:11:11Z"  
Regular Expression: `^([\d]{0-9}[\d]{0-3})?([\d]{0-9})?([\d]{0-9}(\.[\d]{0-9})?)?Z([\+\-]|\+)|([\d]{0-9}[\d]{0-3})?([\d]{0-9})?Z([\d]{0-9})?Z$`
- embargo-description** [Type: PID-InfoType]  
specifies an embargo period together with additional contact information
- date-time** [Type: PID-BasicInfoType]  
Description: combined date and time representations as string. It refers to RFC3339 and ISO 8601 and allows to give just date and combined date and time in UTC, but not week or ordinal date notation. Defined by regular expression.  
Regular Expression: `^([\d]{0-9}(\d{4})?)?([\d]{0-9}[\d]{0-2})?([\d]{0-9}[\d]{0-1})?T([\d]{0-9}[\d]{0-3})?([\d]{0-9}(\.[\d]{0-9})?)?Z([\+\-]|\+)|([\d]{0-9}[\d]{0-3})?([\d]{0-9}[\d]{0-9})?([\d]{0-9})?Z([\d]{0-9})?Z$`
- date-time-rfc3339** [Type: PID-BasicInfoType]  
Description: combined date and time representations as string (RFC3339; ISO 8601), defined by regular expression, because the JSON "date-time"-format is not restrictive enough, allowing "2000-12-88T88:88:88Z".  
Regular Expression: `^([\d]{0-9}(\d{4})?)?([\d]{0-9}[\d]{0-2})?([\d]{0-9}[\d]{0-1})?T([\d]{0-9}[\d]{0-3})?([\d]{0-9}[\d]{0-9})?([\d]{0-9}(\.[\d]{0-9})?)?Z([\+\-]|\+)|([\d]{0-9}[\d]{0-3})?([\d]{0-9}[\d]{0-9})?([\d]{0-9})?Z([\d]{0-9})?Z$`
- full-date** [Type: PID-BasicInfoType]  
Description: combined date and time representations as string (RFC3339; ISO 8601), defined by regular expression, allows "2000-12-01"  
Regular Expression: `^([\d]{0-9}(\d{4})?)?([\d]{0-9}[\d]{0-2})?([\d]{0-9}[\d]{0-1})$`

# Type conformity and validation

- InfoType information has to be particularly reliable, because
  - the functionality of the data services is dependent on a correct preprocessing.
  - this kind of metadata is interpreted by automated services
  - therefore it is necessary to avoid each precondition of human interpretation.
- **Schemas** need to be part of the information type description
  - and have to be defined in a clearly determined and reproducible way.
  - only an automatic process can guarantee this.



Ulrich  
Schwardmann,  
GWDG

DTRs and  
PID Info  
Types

Type  
Hierarchies

Schemas

Type  
Validation

Discussion

# Hierarchies of Types

- Information types are often referring to simpler types:
  - a geolocation contains longitude, given in sexagesimal or decimal form.
  - citation information contains an author, perhaps given by an ID in a certain ID system
- they are eventually based on very basic types
  - determined by regular expressions or other restrictions

## Suggestion:

- Information types are recursively built out of a finite combination of
  - information types and
  - such basic information types
- possible advantage: reuse of schemas



ePIC  
DTR4PIT

Ulrich  
Schwardmann,  
GWDG

DTRs and  
PID Info  
Types

Type  
Hierarchies

Schemas

Type  
Validation

Discussion

# Hierarchies of Types

21.T11148/a77cd6959b4fff9a9c50  
Type Name: time-period  
Type: PID-InfoType



21.T11148/a045f55e2a7fc9d60a5b  
Type Name: date-time  
Type: PID-BasicInfoType



ePIC  
DTR4PIT

Ulrich  
Schwardmann,  
GWDG

DTRs and  
PID Info  
Types

Type  
Hierarchies

Schemas

Type  
Validation

Discussion

# Schemas, how to generate and maintain them?

- common practice: manually build a schema according to a given description
  - a lot of manual work for schema derivation and adaption
  - inconsistencies between description and the schema

automatic schema derivation (in JSON):

- exactly describe the information type dependencies in the type description in the DTR
  - enable as much flexibility in the JSON framework as possible
  - the description of dependencies is a derivation from the canonical type description set
- exploit the hierarchy in an automated process
- basic information types have a (simple) schema as leaves in the dependency graph
- store the schema in the type description of the DTR



ePIC  
DTR4PIT

Ulrich  
Schwardmann,  
GWDG

DTRs and  
PID Info  
Types

Type  
Hierarchies

Schemas

Type  
Validation

Discussion

# Hierarchies of Types

```
{
  "identifier" : "21.T11148/a77cd6959b4fff9a9c50",
  "name" : "time-period",
  "description" : "describes a time period between given begin
every time before end-time or after begin-time. An empty time
standards" : [ {
  "natureOfApplicability" : "depends",
  "name" : "21.T11148/a045f55e2a7fc9d60a5b",
  "issuer" : "DTR"
} ],
"provenance" : {
  "contributors" : [ {
    "identifiedBy" : "Handle",
    "name" : "Ulrich Schwardmann",
    "detail" : "GWDC"
  } ]
},
"creationDate" : "2016-02-23T12:35:41.963Z",
"lastModificationDate" : "2016-07-17T09:34:38.430Z"
},

```

```
"representationsAndSemantics" : [ {
  "expression" : "Measurement Unit",
  "value" : "time",
  "subSchemaRelation" : "denyAdditionalProperties",
  "allowAbbreviatedForm" : "Yes"
} ],
"properties" : [ {
  "name" : "begin-time",
  "identifier" : "21.T11148/a045f55e2a7fc9d60a5b",
  "representationsAndSemantics" : [ {
    "expression" : "Measurement Unit",
    "value" : "time",
    "obligation" : "Optional",
    "repeatable" : "No"
  } ]
}, {
  "name" : "end-time",
  "identifier" : "21.T11148/a045f55e2a7fc9d60a5b",
  "representationsAndSemantics" : [ {
    "expression" : "Measurement Unit",
    "value" : "time",
    "obligation" : "Optional",
    "repeatable" : "No"
  } ]
} ],

```

```
"validationSchema" : "{\n  \"time-period\" : {\n    \"additionalPropert\n  }\n  \"begin-time\" : {\n    \"additionalPropert\n  }\n  \"end-time\" : {\n    \"additionalPropert\n  }\n}\n",
"validationSchema" : "{\n  \"time-period\" : {\n    \"additionalPropert\n  }\n  \"begin-time\" : {\n    \"additionalPropert\n  }\n  \"end-time\" : {\n    \"additionalPropert\n  }\n}\n",

```

```
"representationsAndSemantics" : [ {
  "expression" : "Measurement Unit",
  "value" : "time",
  "subSchemaRelation" : "denyAdditionalProperties",
  "allowAbbreviatedForm" : "Yes"
} ],
"properties" : [ {
  "name" : "begin-time",
  "identifier" : "21.T11148/a045f55e2a7fc9d60a5b",
  "representationsAndSemantics" : [ {
    "expression" : "Measurement Unit",
    "value" : "time",
    "obligation" : "Optional",
    "repeatable" : "No"
  } ]
}, {
  "name" : "end-time",
  "identifier" : "21.T11148/a045f55e2a7fc9d60a5b",
  "representationsAndSemantics" : [ {
    "expression" : "Measurement Unit",
    "value" : "time",
    "obligation" : "Optional",
    "repeatable" : "No"
  } ]
} ],
"validationSchema" : "{\n  \"time-period\" : {\n    \"additionalPropert\n  }\n  \"begin-time\" : {\n    \"additionalPropert\n  }\n  \"end-time\" : {\n    \"additionalPropert\n  }\n}\n",

```

# Validation of instances

- all necessary information for type validation can be found in the PID and the DTR
  - and can be retrieved via the REST API of the Handle System or the DTR
- client function calls like `typeIsValid(PID,typeID)` are provided



ePIC  
DTR4PIT

Ulrich  
Schwardmann,  
GWDG

DTRs and  
PID Info  
Types

Type  
Hierarchies

Schemas

Type  
Validation

Discussion



# Validation of DataCite Mandatory Properties

- all mandatory metadata entries for the DataCite MDS are also defined in JSON in the DTR
- types for all metadata entries for the DataCite MDS will follow
- additionally there is a collective type of these mandatory properties defined in the DTR
- for both metadata instances in JSON and XML exist schema
- there exists a crosswalk from the ePIC JSON to the DataCite XML
  - validation of the JSON type instance guarantees a valid DataCite XML



Ulrich  
Schwardmann,  
GWDG

DTRs and  
PID Info  
Types

Type  
Hierarchies

Schemas

Type  
Validation

Discussion

Conclusion: Types are a great medium to enable more interoperability between PID systems

# Validation of tabular data with the Frictionless Data Schema



ePIC  
DTR4PIT

Ulrich

- Frictionless Data has a hierarchical schema description for necessary data to validate and process tabular data

The screenshot shows the 'Document Repository' interface with a search bar containing 'FLD'. The search results list several schema terms with their descriptions and default values:

- primaryKey-FLD** [Type: PID-InfoType]  
A primary key is a field or set of fields that uniquely identifies each row in the table given by a CSV file.  
Default Value: true
- reference-FLD** [Type: PID-InfoType]  
a reference where entries in a given field (or fields) on this table ('resource' in data package terminology) is a reference to an entry in a field (or fields) on a separate resource.  
Default Value: true
- foreignKeys-FLD** [Type: PID-InfoType]  
A foreign key is a reference where entries in a given field (or fields) on this table ('resource' in data package terminology) is a reference to an entry in a field (or fields) on a separate resource.  
Default Value: true
- fields-FLD** [Type: PID-InfoType]  
fields is an ordered list of field descriptors one for each field (column) in the table  
Default Value: true
- CSV-table-description-FLD** [Type: PID-InfoType]  
A CSV table description intends to describe the structure of tables consisting of a set of rows. Each row has a set of fields (columns). We usually expect that each row has the same set of fields and thus we can talk about the fields for the table as a whole. It describes the delimiter in the CSV file between columns as well as the constraints of the columns and primary keys and similar.  
Default Value: true
- field-FLD** [Type: PID-InfoType]  
A field descriptor of Frictionless Data is a simple JSON hash that describes a single field. The descriptor provides additional human-readable documentation for a field, as well as additional information that may be used to validate the field or create a user interface for data entry.
- csvddf-FLD** [Type: PID-InfoType]  
CSV Dialect Description Format defines a simple JSON format to describe the various dialects of CSV files; it aims to deal with a reasonably large subset of the features which differ between dialects (terminator strings, quoting rules, escape rules, etc), and roughly to describe the union of the capabilities of Python's csv module, Ruby's CSV module, and the MySQL and PostgreSQL load facilities of the *line of writing* (Eckstein 2012). The goal of the "dialect" member of the JSON dictionary is intended to be used close to the

# Validation of tabular data with the Frictionless Data Schema

- The FDL types together with corresponding resulting schemas are described in DTR

```
"properties" : {
  "CSV-table-description-FLD" : {
    "additionalProperties" : false,
    "description" : "CSV-table-description-FLD@21.T11148_66ee7993765837104ce3",
    "properties" : {
      "CSVDDF" : {
        "$ref" : "#/definitions/21.T11148_bbbe2b8ca636e95eaffc"
      },
      "fields" : {
        "$ref" : "#/definitions/21.T11148_39c5c9e5b9a54b398562"
      },
      "foreignKeys" : {
        "$ref" : "#/definitions/21.T11148_071c82c85172bed4a8a5"
      },
      "primaryKey" : {
        "$ref" : "#/definitions/21.T11148_17ce40f9612c494cc3c3"
      }
    },
    "required" : [
      "fields",
      "CSVDDF"
    ],
    "type" : "object"
  }
},
"required" : [
  "CSV-table-description-FLD"
],
"type" : "object"
}
```



ePIC  
DTR4PIT

Ulrich  
Schwardmann,  
GWDG

DTRs and  
PID Info  
Types

Type  
Hierarchies

Schemas

Type  
Validation

Discussion

# Validation of tabular data

Example: Validate the CSV file at  
<http://techslides.com/demos/country-capitals.csv>

- columns are country code, city name, latitude, longitude, region-code, continent-names
- for these columns exist (candidate) InfoTypes
- Validation by a simple python script
  - gets an array of the PITs of the column types
  - retrieves the CSV file above
  - gets the names and the schemas for the column types from the DTR
    - using the client library from above
  - validates the column entries with the schemas
  - and outputs each row with the offended column entry



ePIC  
DTR4PIT

Ulrich  
Schwardmann,  
GWDG

DTRs and  
PID Info  
Types

Type  
Hierarchies

Schemas

Type  
Validation

Discussion

# Validation of tabular data

Example: Validate the CSV file at  
<http://techslides.com/demos/country-capitals.csv>

- columns are country code, city name, latitude, longitude, region-code, continent-names
- for these columns exist (candidate) InfoTypes
- Validation by a simple python script
  - gets an array of the PITs of the column types
  - retrieves the CSV file above
  - gets the names and the schemas for the column types from the DTR
    - using the client library from above
  - validates the column entries with the schemas
  - and outputs each row with the offended column entry



ePIC  
DTR4PIT

Ulrich  
Schwardmann,  
GWDG

DTRs and  
PID Info  
Types

Type  
Hierarchies

Schemas

Type  
Validation

Discussion

# Validation of tabular data



ePIC  
DTR4PIT

Ulrich  
Schwardmann,  
GWDG

DTRs and  
PID Info  
Types

Type  
Hierarchies

Schemas

Type  
Validation

Discussion

```
uschwar1 : bash - Konsole <5>
Datei Bearbeiten Ansicht Lesezeichen Einstellungen Hilfe
uschwar1@pcscw:~> python csv-type-validator.py '['21.
T11148/f1627ce85386d8d75078", "21.T11148/f1627ce85386
d8d75078", "21.T11148/5fccccdf1d079c4a85c9", "21.T111
48/d2a773ae817d7d07c19d", "21.T11148/16b9945aca671268
bb5e", "21.T11148/7144a1b3d87e01a52d86"]' country-cap
itals.csv

ERROR at line 0002 item 4: Somaliland,Hargeisa,9.55,4
4.050000, NULL, Africa
ERROR at line 0027 item 2: Bangladesh,Dhaka,90.400000
,23.716666666666665,BD,Asia
ERROR at line 0229 item 2: United States,Washington,
D.C.,38.883333,-77.000000,US,Central America
ERROR at line 0242 item 4: Northern Cyprus,North Nico
sia,35.183333,33.366667, NULL, Europe
```

# Endorsement and Deprecation of Types

- registration of PID InfoTypes is an endorsement process
  - currently all registered PIT are **in preparation**
  - after their description is sufficiently settled they become **candidates**
  - after there is no change request anymore they become **approved** and will be frozen
  - new version is necessary for the changement of approved types
    - in this case the former type becomes **deprecated**
- the endorsement of PID InfoTypes needs a **reviewing board**



ePIC  
DTR4PIT

Ulrich  
Schwardmann,  
GWDG

DTRs and  
PID Info  
Types

Type  
Hierarchies

Schemas

Type  
Validation

Discussion

# Questions and Discussion



ePIC  
DTR4PIT

Ulrich  
Schwardmann,  
GWDG

**Thanks** for your attention !

View also the ePIC Webpage:

`http://pidconsortium.eu`

DTRs and  
PID Info  
Types

Type  
Hierarchies

Schemas

Type  
Validation

Discussion