THIS IS THE WORKING DOCUMENT; COMMENTS WELCOME.

The working copy of this document can be found at https://docs.google.com/document/d/1PmqYur26JnnAytC4n4_zFwY1efggvL09YTU2zgF9Gkk/edit#heading=h.mfbbtg2a3oen

The IUPAC FAIRSpec Metadata Object Model Specification, Version 0.0.4

Mark Archibald, Ian Bruno, Stuart J. Chalk, Robert M. Hanson, Antony N. Davies, Damien Jeannerat, Robert J. Lancashire, Jeff Lang, Henry S. Rzepa

[draft version 2022.11.25]

TODO:
- align field names with Java classes for derived classes

partial history:
2022.11.25 BH adds IFDParameter; changes "." to "_" in terminal names
2022.06.08 BH adds note in subclassing about not limiting superclasses
2022.04.20 BH adds *.note, *.description to IFDObject
2022.04.12 BH uses id rather than label for collection references; adds IFDCollection.itemType
2022.04.11 BH preliminary field descriptions;
2021.06.26 BH created

Index

This Document

This document describes a preliminary specification for data management of spectroscopic data in the area of chemistry that allows for a seamless process from generation of experimental data through analysis to publication and public archiving. Specifically, it describes the abstract model that underlies the metadata associated with an IUPAC FAIRData collection, including the IUPAC FAIRSpec extension of that model.

The GitHub repository for this project is at https://github.com/IUPAC/IUPAC-FAIRSpec, and a set of sample datasets with associated IUPAC FAIRSpec Finding Aids can be found in https://chemapps.stolaf.edu/iupac/ifd2 with an interactive demo at https://chemapps.stolaf.edu/iupac/demo2/demo.htm.

A presentation of the principles associated with this work are elaborated upon in the *Pure and Applied Chemistry* article *IUPAC Specification for the FAIR Management of Spectroscopic Data in Chemistry (IUPAC FAIRSpec) - Guiding Principles* (accepted Mar 13, 2022)

The goal of the IUPAC Project 2019-031-1-024 is to enable a standardized way of managing spectroscopic data digital collections. This document sets forth a set of standards for the description and cataloging of data and its associated metadata in ways that are practical, relatively simple to implement, modular, intuitive, easily extended, and flexible in terms of requirements for the data itself as well as format the metadata can be expressed in.

The scope of the project is *spectroscopic* data within a *chemical* context, but what is described here is fully extensible to any sort of data, within any context, and is already being proposed in the area of materials science.

The proposed standards involve several aspects:

- a set of principles underlying what we mean by "FAIR" in relation to spectroscopic data.
- a detailed **metadata object model** for describing the contents of a "spectroscopic data collection" in terms of objects and relationships of objects as described by metadata,
- a standard for **describing properties** of digital objects within the metadata records of the finding aid,
- a standard for the **serialization of the finding aid** for a collection,
- a recommendation for the **organization** of digital objects within a collection,
- a proposal for and demonstration of methods of **data and metadata extraction** and the generation of IUPAC FAIRSpec Finding Aids.

This document focuses on the second of these bullet points, specifically describing the eleven metadata classes comprising the core of the metadata object model, along with twelve standard digital object-related metadata classes and several additional classes that are suggested for general use and are examples of how the model can be extended to specific applications.

1. IUPAC FAIRSpec Principles

These Principles are elaborated upon in the *Pure and Applied Chemistry* article *IUPAC Specification for the FAIR Management of Spectroscopic Data in Chemistry (IUPAC FAIRSpec) - Guiding Principles* (accepted Mar 13, 2022). Only a brief summary is given here in order to provide the rationale for the data model.

Spectroscopic data in the area of chemistry are intimately connected to chemical structure. The whole purpose of carrying out spectroscopic measurements is often to discover the identity of pure compounds, to identify and quantitate the relative amounts of chemical impurities in a mixture, and to determine the structure of new compounds.

In addition, there are contexts where it is more appropriate to refer to spectroscopic data in relation to "samples." We want to know the identity of a sample, but we don't yet know its chemical structure. Or, perhaps we are working with a material that is not characterizable *per se* as a chemical compound. These standards cover such cases as well.

Spectroscopy datasets can be complex. We make the distinction here between *Digital Entities* (sequences of bytes, for example "files"), which, through metadata association, become *Digital Objects*.[def reference] A key concept in the IUPAC FAIRSpec data model is the ***IUPAC FAIRSpec Digital Collection***, which uses metadata to make useful connections between digital objects and allows for a variety of "representations" of those objects (Figure 1.1).

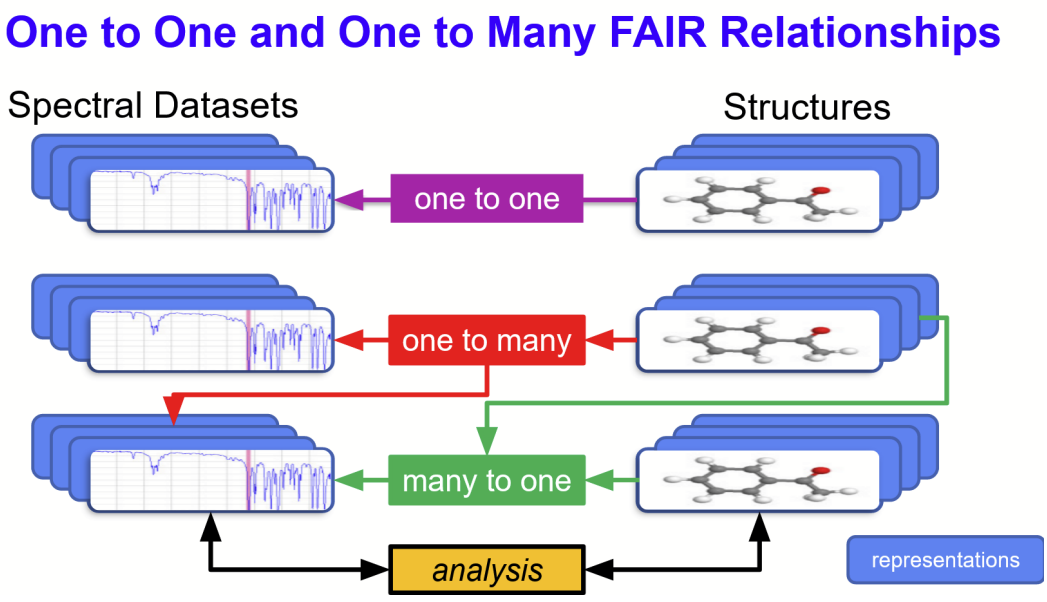


Figure 1.1. Key relationships among FAIRSpec Digital Objects.

Thus, the IUPAC FAIRSpec Standard data model is not about standardizing instrument data file formats (though it benefits from that) or requiring specific representations (though it might make specific recommendations in this regard). Rather, it is about providing a baseline common ground for the storage, transmission, and description of the contents of a digital collection derived from spectroscopic measurements or calculations. The ultimate goal is to be able to provide concise

descriptions of complex data sets associated with manuscripts, ongoing laboratory work, and teaching, that can be findable, accessible, interoperable, and reusable by machines and humans alike.

The following principles underlie our development of the IUPAC FAIRSpec standards:

1. FAIR Management of data should be an ongoing concern.

A. FAIR management of data must be an explicit part of research culture.
B. FAIR management of data should be of intrinsic value.
C. Good data management requires distributed curation.
D. Experimental work is by nature iterative.

2. Context is important.

A. Digital objects are generally part of a collection.
B. Chemical properties are related to chemical structure.
C. Data relationships are diverse and develop over time.
D. FAIR management of data should allow for validation.

3. FAIR management of data requires curation.

A. Data reuse relies upon practical findability.
B. Data has to be organized to be accessible.
C. Data interoperability requires well-designed metadata.
D. Value is in the eye of the reuser.

4. Metadata must be standardized and registered.

A. Register key metadata.
B. Assign a variety of persistent identifiers.
C. Enable metadata crosswalks.
D. Allow for value-added benefits.

5. FAIR data management standards should be modular, extensible, and flexible

A. Modularity allows specialization.
B. Allow for future needs.
C. Respect format and implementation diversity.
D. All data formats should be valued.

2. The IUPAC FAIRData Metadata Object Model

What follows is a detailed discussion of the metadata object model underlying the IUPAC FAIRSpec specification. We refer to this model as the "IUPAC FAIRData" model because it is a general model that is not specific in any way to spectroscopy. "IUPAC FAIRSpec" refers to the IUPAC FAIRSpec Extension of the IUPAC FAIRData model. The point here being that the model is easily extensible, and IUPAC FAIRSpec is just one example of how that can be accomplished.

The discussion is framed in the language of object-oriented programming in order to more clearly define the relational aspects, scope, and limitations of the model.

2.1 Definitions

For readers not familiar with object-oriented programming, we offer this brief set of definitions:

- *object*

  A construct having properties (or *fields*), which may include its relationships to other objects. An object may be *representable* in the form of bits and bytes (a *digital object*) or it may be an abstract set of properties(a *metadata object*).
- *class*

  An abstract description of an object's type, expressing its capabilities and limitations. We say that an object is an "instance of its class", or that "instantiation of a class" creates an object. All IUPAC FAIRData Metadata Object Model classes start with "IFD".

- *subclass* and *superclass*

  The "child" of a *superclass* is a *subclass.* We say that a subclass *extends* its superclass so that it can "inherit" the superclass's properties. This efficient design allows general characteristics to be described in one class (a superclass) and propagated to a whole family of other classes (its subclasses). Subclassing allows for expansion and customization of the model without the need to change the core classes. So, for example, we will see that the core aspect of an *IFDCollection* (that it is a list of IFDObjects) is also true of its subclass IFDStructureCollection, but in this case, the objects are limited specifically to being IFDStructure objects. Subclassing does *not* allow for limiting the capabilities of a superclass.

- *abstract class*

  An abstract class is a class that can only be a superclass and can never be instantiated itself. It is basically just a template that provides common properties for its various subclasses. In the IUPAC FAIRData Metadata Object Model, only the classes IFDObject, IFDRepresentation, IFDRepresentableObject, IFDCollection, IFDDataObject, and IFDDataObjectRepresentation are abstract classes. This ensures that, in each case, an object of one of these types must be more specifically defined. For example, while IFDObject could be just about anything, IFDSample and

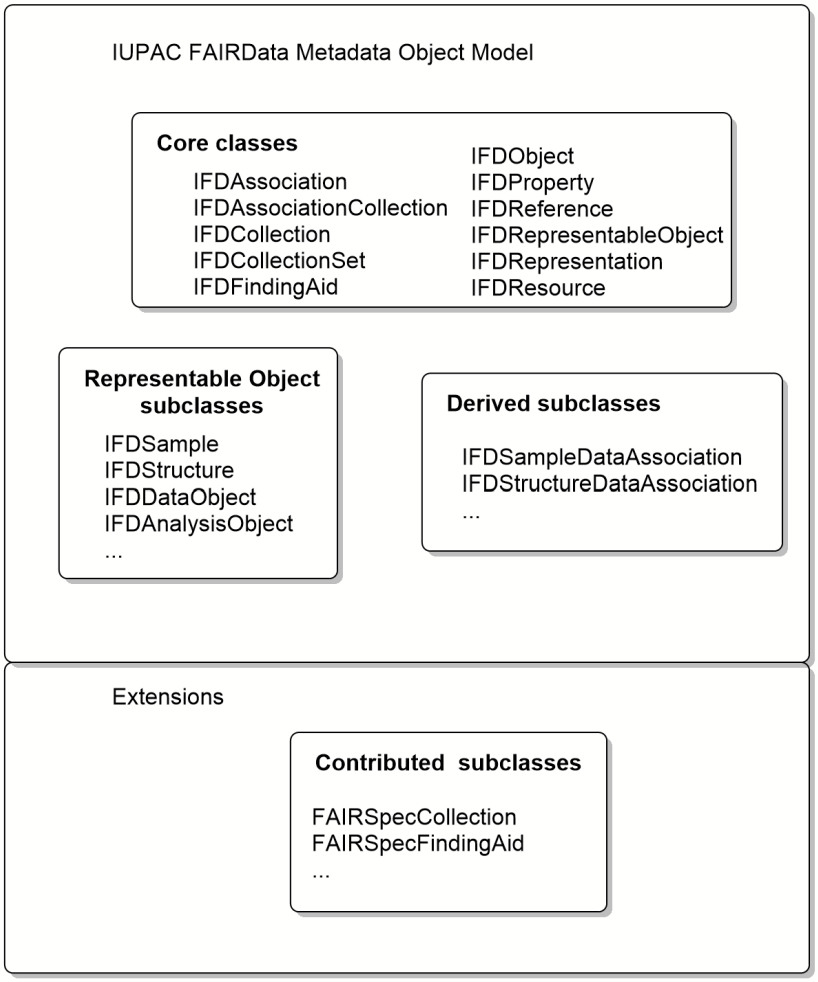
IFDStructure are subclasses of type IFDRepresentableObject (itself a subclass of IFDObject) that have distinct representations characteristic of their type.

- *field*

  An attribute of a class that has a data type, which is itself a reference to a class (String, Integer, Double, List, Map, etc. – See Appendix A). Data types can be qualified. For example, a List of String will be expressed here as List<String>. Fields that are arrays are indicated with [ ] after their name – *items[ ]*, for example.

2.2 Overview

2.2.1 The core model



An overview of the classes in the model is shown in Figure XX. The model includes three primary groups of classes:

Core classes are classes that describe the basic elements of the model – metadata objects and their associations, digital representations and collections of digital objects, metadata structures including properties, references, and resource information that characterize these objects. An instance of IFDFindingAid class maintains a single IFDCollectionSet, which itself is a collection of collections of all of the other sorts of metadata objects.

Representable Object subclasses are specialized metadata objects that point to and describe the digital object of the actual digital collection – objects relating to physical samples, chemical structures, various digital objects acquired from experimental or computational work (collectively referred to hear as "data objects"), and post-acquisition analyses objects.

Derived subclasses are classes that associate two or more different types of representable objects – for example, a physical sample with one or more chemical structures, chemical structures with data objects such as NMR spectra, and detailed analyses that provide justification for those associations.

2.2.2 Extending the model – contributed subclasses and metadata specifics

Importantly, the model is designed to be customized in ways that go well beyond generic data. Any of the classes of the model or its extensions can be subclassed further to provide a variety of additional object types, allowing for collections of different sorts. All that is required by the model is that the model classes themselves are still identifiable, and that the subclassing itself is well described. Customization can be accomplished in any combination of the following three ways:

Contributed subclasses can be designed and added to the model. For example, a FAIRSpecFindingAid object (a subclass of IFDFindingAid) describes a FAIRSpecCollection (a subclass of IFDCollectionSet). The process of subclassing introduces specialized metadata properties and representations that are specific in this case to the area of spectroscopy.
When serialized in a machine-readable form such as JSON or XML, the FAIRSpec Finding Aid produces the digital object we refer to as the *IUPAC FAIRSpec Finding Aid*, which describes the contents of an *IUPAC FAIRSpec Digital Collection*.

Elaboration of metadata elements provides another means of customization. Note that the model itself is largely independent of the details and format of the descriptive metadata key/value properties themselves that we traditionally associate with "metadata". This is by design. Though there are a few basic rules in the model as to how these keys in particular are constructed (see ifd.properties), the details of those keys and their associated values are left to extension contributors.

Addition of nonstandard parameters is also allowed. Nothing in the specification requires the *exclusive* use of IUPAC FAIRData metadata key/value pairs. The finding aid has two property-like categories. *Properties* are metadata elements that have standardized IUPAC FAIRData (or its extensions') keys and allowed values. *Parameters* are metadata of completely free format, or of a format described specifically by the implementor or by other standards.

2.3 Core Classes

class IFDAssociation extends IFDCollection
class IFDAssociationCollection extends IFDCollection
class IFDCollection extends IFDObject
class IFDCollectionSet extends IFDCollection
class IFDFindingAid extends IFDObject
class IFDObject extends ArrayList
class IFDParameter
class IFDProperty
class IFDReference
class IFDResource
class IFDRepresentableObject
class IFDRepresentation

The eleven core model classes are fundamental classes that have no specific reference to spectroscopy. Thus, the core specification easily could be extended to non-spectroscopic FAIR data management applications.

2.3.1 IFDObject (abstract)

IFDObject is a superclass for nearly all IUPAC FAIRSpec Data Model objects.

- IFDObject
  - IFDCollection
    - IFDAssociation
    - IFDAssociationCollection
    - IFDCollectionSet
    - IFDSampleCollection
    - IFDStructureCollection
      …(several more)
  - IFDRepresentableObject
    - IFDAnalysisObject
    - IFDDataObject
    - IFDSample
    - IFDStructure
  - IFDFindingAid

Figure xxx. Subclasses of IFDObject in the IUPAC FAIRSpec Metadata Object Model.

org.iupac.fairdata.core.IFDObject		
field name	**data type**	**description**
description	String	an optional
label	String	an optional short descriptive but not necessarily unique label for this object

id	String	an optional unique identifier for this object; if present, must be unique within this finding aid
ifdType	String	required; the class name of this object, such as *org.iupac.fairdata.sample.IFDSample*
ifdTypeExtends	String	required for non-core classes; a semicolon-separated list of names of classes in the superclass hierarchy of this class, up to the first class name that is of the form *org.iupac.fairdata.core.** (only present for non-core classes)
note	String	an optional free text note, possibly relating to issues in the production of the finding aid or collection; not intended for general metadata indexing
parameters	Map<String, String>	an optional set of key/value pairs that are not part of the IUPAC FAIRData standard or its extension. Characters of the keys must be one of [A-Za-z0-9_] or the space character and must be trimmed of any space character.
properties	Map<String, String>	an optional set of key/value pairs where the key is a fully qualified IUPAC FAIRData property name starting with "IFD." or just its associated short (underscore-separated) name, if the propertyPrefix field is also present. The value must be of the form specified by this standard or its extension
propertyPrefix	String	optional; a prefix starting with "IFD." that should be prepended along with a "." to each property key not already starting with "IFD." in order to produce a fully qualified IUPAC FAIRData property name as specified by this standard or its extension

Note that IFDObject.properties and IFDObject.parameters both declare metadata properties as key:value pairs. The properties field is reserved for properties that are described by the standard, while the parameters field holds properties that are not (yet) part of the standard. This allows implementers to introduce properties that are important to their work without breaking the standard. Future versions of the standard may include those properties using standardized syntax, adding information about units and type. In addition, because IFDObject extends ArrayList (a Java class maintaining an array of objects that can be added to dynamically), all IFDObjects are inherently list-like.

IFDObject has three direct subclasses: IFDRepresentableObject, IFDCollection, and IFDFindingAid. An IFDRepresentableObject is a list of IFDRepresentation items, while an IFDCollection is a list of IFDRepresentableObject items. The IFDFindingAid is a special object that is not itself a collection but holds the IFDCollectionSet. These three subclasses are described in detail in section *2.3 Core Classes*.

2.3.2 IFDRepresentation (abstract)

IFDRepresentation is an abstract class with instances that either reference actual digital objects or contain byte sequence or string data themselves. These may be images, spreadsheets, complex analyses, discipline-specific or vendor-specific file formats – anything format that data (in the broad sense, including chemical structure "data") can be represented. All actual digital objects in an IUPAC FAIRData Collection will be represented in the model as an IFDRepresentation of one type or another and serialized as part of a List<IFDRepresentation>, where each representation in the list is in the form of a Map<String, String>. Note that IFDRepresentation objects themselves are not IFDObject objects and are not list-like. As an abstract class, IFDRepresentation can only be "realized" as a more specific sort of representation:

- IFDRepresentation
    - IFDAnalysisObjectRepresentation
    - IFDDataObjectRepresentation
    - IFDSampleRepresentation
    - IFDStructureRepresentation

Figure xxx. Subclasses of IFDRepresentation differentiates distinctly different types of representation.

IFDRepresentation.data could refer to any digital entity. Subclasses of IFDRepresentation provide a way of ensuring that the correct general type of digital object in an IUPAC FAIRData Collection corresponds with a given IFDRepresentableObject. For example, chemical structure-related metadata objects (IFDStructure objects) must be lists of structure representations (IFDStructureRepresentation objects, such as MOL files). That is, IFDStructure (which is a List) extends IFDRepresentableObject<IFDStructureRepresentation>.

org.iupac.fairdata.core.IFDRepresentation		
field name	**data type**	**description**
data	String	when the representation itself is suitable (in the opinion of the implementer), the representation can be serialized directly within the finding aid as a String; values that start with the prefix ";base64," are to be interpreted as base64-encoded bytes as specified in RFC 4648; when the data field is present, the ref field should not be present in the serialization, or should have the literal value "null" (without quotes)
ifdType	String	the object type of the form *IFD.representation…* declared in the extension of the specification
key	String	the name of this representation, starting with "IFD.representation." and continuing with a dot-separated set of qualifiers that consist only of lower-case or digit [a-z0-9] characters; for example: *IFD.representation.dataobject.fairspec.nmr.image*

len	Integer	the byte length of the digital object associated with this representation
mediaType	String	one or more semicolon-separated media types describing the media type of this digital object, if known and relevant
ref	IFDReference	the machine-actionable reference to the digital object associated with this reference, or null when the data field is non-null.

2.3.3 IFDParameter

IFDParameter is the storage class for non-standardized metadata key/value pairs for all IFDObjects. An IFDParameters can use one name to refer to a single value or a list of values.

org.iupac.fairdata.core.IFDProperty		
field name	**data type**	**description**
name	String	The name of this parameter consisting of a dot-separated sequence referring to a class followed by a dot and a sequence of lower-case or digit or underscore [a-z0-9] characters; for example: *IFD.property.collectionset.source_data_uri*
value	(variable)	the value of this metadata object, serialized as a number, boolean, string, or a list of such values.

2.3.4 IFDProperty

IFDProperty is the storage class for standardized metadata key/value pairs for all IFDObjects. This class can be serialized in-line as a key:value pair within a Map<String,String> serialization.

org.iupac.fairdata.core.IFDProperty		
field name	**data type**	**description**
(key)	String	The name of this property, starting with "IFD.property." and continuing with a dot-separated set of qualifiers that consist only of lower-case or digit [a-z0-9] characters; for example: *IFD.property.dataobject.spec.nmr.expt.freq1*

(value)	(variable)	the value of this metadata object, serialized as the data type and declared by the standard or its extension, in the units specified by the standard or its extension

Properties, in principle, may have specified data types (String, Integer, Integer, Float64, …), units (Hz, MHz, Kelvin, …), values, value ranges ([1…]), or value formats (1H, 13C, …) that are specified in the standard or its extension. Properties may be serialized as key:value pairs within a map rather than as explicit declarations such as *key=... value=...*.

2.3.5 IFDCollection

IFDCollection is an abstract class designed to hold a set of objects of the same class or superclass. IFDCollection objects are not themselves IFDRepresentableObjects and thus have no digital representation themselves. Note that IFDCollection objects may be collections of IFDCollection objects.This allows for a nesting of collections in meaningful ways. For example, the subclass IFDStructureDataAssociation (a subclass of IFDCollection) maintains two specific collections: one for structures, and one for data objects.

org.iupac.fairdata.core.IFDCollection inherits fields of IFDObject		
field name	**data type**	**description**
items[ ]	(variable)	the items in the collection
itemType	String	class name of all items (present only for where all items are the same type, allowing for individual items to not declare their itemType)
itemTypeExtends	String	a semicolon-separated list of names of classes in the superclass hierarchy of this class, up to the first class name that is of the form *org.iupac.fairdata.core.** (present only for where all items are the same type, allowing for individual items to not declare their itemType)

To be sure, an IFDCollection could represent a digital object in the form of a zip file (and usually does). Nonetheless, this does not make it "representable" in the sense that it is likely to have multiple distinctly different representations that characterize an IFDRepresentableObject. Instead, we would say that an IFDCollection is serializable, meaning it can be converted into string form.

2.3.6 IFDAssociation

IFDAssociation is a class that is a specialized "collection of collections." The collection is intended to be "immutable" in the sense that some initial set of collections are associated, and then the collection is fixed. The number of items in the association is entirely customizable.

org.iupac.fairdata.core.IFDAssociation inherits fields of IFDCollection and IFDObject		
field name	**data type**	**description**
items[ ][ ]	Integer	(from IFDCollection) each item is an array of arrays of integers, one array for each of the components of the association; numbers are indexes into the specified named collections in the finding aid's collections set.

IFDAssociation is particularly used for metadata that relates one object to another – a sample to a set of spectra; a set of spectra to a structure. An IFDAssociation in its simplest serialized form is just a set of pointers to representable objects in other collections of the FAIRData Finding Aid. IFDAssociations are serialized as the items of an IFDAssociationCollection. Thus, in serialized form, an association of samples and their spectra might appear as shown in bold, below:

```
"sample-spectra-associations":{
        "type":"org.iupac.fairdata.derived.IFDSampleDataAssociationCollection",
        "typeExtends":"org.iupac.fairdata.core.IFDAssociationCollection",
        "name":"sample-spectra-associations",
        "id":"6",
        "collections":["samples","spectra"],
        "items":[
         {"type":"org.iupac.fairdata.derived.IFDSampleDataAssociation",
          "typeExtends":"org.iupac.fairdata.core.IFDAssociation",
          "items":[[0], [0,1,84]]
         },
        …]
}
```

Thus, associations essentially provide the "edges" to a graph having IFDRepresentableObject nodes.

2.3.7 IFDAssociationCollection (abstract)

IFDAssociationCollection objects are lists of IFDAssociation objects.

org.iupac.fairdata.core.IFDAssociationCollection inherits fields of IFDCollection and IFDObject		
field name	**data type**	**description**
collections[ ]	String	an array of the names of the finding aid's collections for which items are being associated, in the same order as the arrays in the IFDAssociation items
items[ ]	IFDAssociation	the associations in this collection

2.3.8 IFDRepresentableObject (abstract)

IFDRepresentableObject maintains a set of metadata relating to different ways a sample or structure or data object or analysis object might be represented. A class extending IFDRepresentableObject is expected to maintain a list of one or more distinctly different digital representations (byte sequences) that amount to more than just metadata.

org.iupac.fairdata.core.IFDRepresentableObject inherits fields of IFDObject		
field name	**data type**	**description**
representations[ ]	IFDRepresentation	an array of representations

Note that there is no mechanism within the model for these classes to refer to each other. This is a key aspect of the IUPAC FAIRSpec Data Model. The model is based on the idea of a collection of independent objects, and it is the IFDAssociation class that accomplishes this referencing. This allows for a relatively simple and flexible packaging of objects, with their relationships identified only in key metadata resources, separately from the objects themselves.

2.3.9 IFDReference

IFDReference is the storage class connecting IFDRepresentations to their original source Digital Objects and their unique local name within an IUPAC FAIRData Collection.

org.iupac.fairdata.core.IFDReference		
field name	**data type**	**description**
localPath	String	the name of the digital object pointed to by this reference; "local" only in the sense that it is local to the path defined by the resource of its associated IFDRepresentation
originPath	String	the path within the originating digital aggregation, if applicable
type	String	the class name of this reference
typeExtends	String	a semicolon-separated list of names of classes in the superclass hierarchy of this class, up to org.iupac.fairdata.core.IFDReference

When *origin* is a ZIP file entry, *origin* will contain a "pipe" vertical line symbol "|" to indicate "a file within a zip file." Directories are represented with the standard forward slash. Thus, *origin* might be something like

1.zip|1_13C/pdata/1/thumb.png

Note that this notation allows for resources that are from zip files that are themselves found within zip files.

Generally speaking, the localName will not be part of a zip file (other than perhaps the zip file used for the collection's delivery), and may even have no hierarchical directory structure, either. For example:

1.zip_1_13C_pdata_1_thumb.png

Note that an IFDRepresentation may or may not have an origin object, and it may or may not have a local path. The lack of an origin indicates that an extractor has created this representation during extraction. For example, using just a SMILES string, it has created a 2D MOL file, a 3D MOL file, InChI, InChIKey, etc. The lack of a localName indicates that a representation's byte data are being saved within the IFDRepresentation object itself, not separately.

2.3.10 IFDResource

IFDResource is a general-purpose structure for holding the byte length and reference for an external resource. It is specifically used in the description of data sources from which an IUPAC FAIRData Collection is derived.

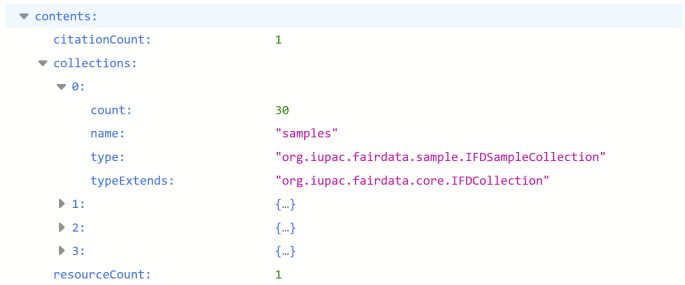
Fields of org.iupac.fairdata.core.IFDResource

field name	data type	description
id	String	a unique identifier of this resource in the finding aid's list of resources
len	Integer	the byte length of this resource
ref	String	the URL of this resource, preferably a persistent identifier
type	String	the class name of this resource
typeExtends	String	a semicolon-separated list of names of classes in the superclass hierarchy of this class, up to org.iupac.fairdata.core.IFDResource

2.3.11 IFDFindingAid and IFDCollectionSet

IFDFindingAid and **IFDColletionSet** constitute the defining metadata objects for an IUPAC FAIRSpec collection and its finding aid. The finding aid provides metadata relating to the entire collection. For example, details about a publication or thesis and source of data. In addition, it maintains one IFDCollectionSet. This collection itself is just a set of collections, consisting of a list of IFDRepresentableObject and IFDAssociation collections and associated metadata relating to the collection as a whole.

org.iupac.fairdata.core.IFDFindingAid inherits fields from IFDObject		
field name	**data type**	**description**
(citations[ ])	*Map<String,?>*	*(currently only part of FAIRSpecFindingAid)*
collectionSet	IFDCollectionSet	The set of collections in the IUPAC FAIRData collection described by this IUPAC FAIRData Finding Aid; this field is the serialized form of a single IFDCollectionSet
contents	Map<String, ?>	a map outlining the contents of this finding aid, format specified by the implementor

		
created	String	a timestamp indicating when this finding aid was created, in the ISO-8601 UTC format *yyyy-MM-ddTHH:mmZ*
createdBy	String	an identifier of the software used to generate this finding aid
resources[ ]	IFDResource	a list of originating resources for this finding aid's collection
version	String	a semicolon-separated list of one or more relevant version identifiers

Note that the *contents* guide is for human readability and convenience only. Nothing in this section is essential to the finding aid itself or standardized for machine readability. Some implementations may provide only this section as the result of a preview query prior rather than replying with a fully elaborated finding aid.

org.iupac.fairdata.core.IFDCollectionSet inherits fields of IFDCollection and IFDObject		
field name	**data type**	**description**
items	IFDCollection	an array of top-level IFDCollections

2.4 Primary subclasses

class IFDSample extends IFDRepresentableObject
class IFDSampleCollection extends IFDCollection
class IFDSampleRepresentation extends IFDRepresentation

class IFDStructure extends IFDRepresentableObject
class IFDStructureCollection extends IFDCollection
class IFDStructureRepresentation extends IFDRepresentation

class IFDDataObject extends IFDRepresentableObject
class IFDDataObjectCollection extends IFDCollection
class IFDDataObjectRepresentation extends IFDRepresentation

class IFDAnalysisObject extends IFDRepresentableObject
class IFDAnalysisObjectCollection extends IFDCollection
class IFDAnalysisRepresentation extends IFDRepresentation

2.4.1 IFDSample, IFDStructure, IFDDataObject (abstract), and IFDAnalysisObject

IFDSample corresponds to a specific physical sample that may or may not (yet) have a chemical structure, spectroscopic data, or representations associated with it. An IFDSampleRepresentation could include a photographic image of the solid, for example. Its properties are minimal, including just at this time just a unique id and a label (such as a laboratory notebook reference).

IFDStructure is a class that corresponds to a chemically-related structural object, which may have several representations in its list, such as a 2D or 3D MOL file, an InChI, one or more chemical names, one or more SMILES strings, or even just a PNG image of a drawn structure. Each of these representations serves a purpose. Some are more "interoperable" than others, but each in its own way may be more useful in a given context. An implementer would have the option to include these representations as data within the finding aid or as an IFDRepresentation reference to a resource within the collection.

IFDDataObject is an abstract class that provides a data structure for maintaining a list of one or more IFDDataObjectRepresentation instances -- what a scientist would call their "data" -- such as a full vendor experiment dataset (a Bruker NMR experiment), a PNG image of a spectrum, a JCAMP-DX or nmrML file, or a peaklist. The model is fully extendable to cover a wide variety of data formats and experimental or simulated techniques.

IFDAnalysisObject instances maintain a list of one or more representations of a relatively detailed post-acquisition analysis. The object provides some sort of description of the rationale for why one particular spectrum, for example, is reasonably associated with one particular structure. The analysis itself is not the structure or the spectrum. This object would normally be part of an IFDAnalysisObjectCollection that is connected to one IFDStructureCollection and one IFDDataObjectCollection via an IFDStructureDataAnalysis metadata object..

org.iupac.fairdata.dataobject.IFDAnalysisObject
org.iupac.fairdata.dataobject.IFDDataObject
org.iupac.fairdata.sample.IFDSample
org.iupac.fairdata.structure.IFDStructure

inherits fields of IFDRepresentableObject and IFDObject
adds no additional fields

2.4.2 IFDSampleCollection, IFDStructureCollection, IFDDataObjectCollection, and IFDAnalysisObjectCollection

These classes group objects of the same general type – sample, structure, data, or analysis, respectively. Each collects distinctly different metadata relating to samples, structures, data, and analysis, respectively.

org.iupac.fairdata.analysisobject.IFDAnalysisObjectCollection
org.iupac.fairdata.dataobject.IFDDataObjectCollection
org.iupac.fairdata.sample.IFDSampleCollection
org.iupac.fairdata.structure.IFDStructureCollection

inherits fields of IFDCollection IFDObject
adds no additional fields

2.4.3 IFDSampleRepresentation, IFDStructureRepresentation, IFDDataObjectRepresentation (abstract), and IFDAnalysisObjectRepresentation

These classes refer to actual digital objects within an IUPAC FAIRData Collection. Subclassing of IFDRepresentation ensures that the representations of an object match.

org.iupac.fairdata.analysisobject.IFDAnalysisObjectRepresentation
org.iupac.fairdata.dataobject.IFDDataObjectRepresentation
org.iupac.fairdata.sample.IFDSampleRepresentation
org.iupac.fairdata.structure.IFDStructureRepresentation

inherits fields of IFDRepresentation
adds no additional fields

2.5 Derived subclasses

These classes listed in this section supply the connecting links among samples, data objects, chemical structures, and post-acquisition data analysis. Whereas an association can have any (fixed) number of elements (in the form of IFDCollection objects), the associations presented here involve only two or three elements. Only representative examples of these classes are described below.

2.5.1 two-element IFDSample-IFDStructure-IFDDataObject associations

class IFDSampleDataAssociation extends IFDAssociation
class IFDSampleDataAssociationCollection extends IFDAssociationCollection

class IFDStructureDataAssociation extends IFDAssociation
class IFDStructureDataAssociationCollection extends IFDAssociationCollection

class IFDSampleStructureAssociation extends IFDAssociation
class IFDSampleStructureAssociationCollection extends IFDAssociationCollection

2.5.1.1 IFDSampleDataAssociation

This class links an experimental sample with its associated data without any explicit association with a chemical structure. It contains two collections, an IFDSampleCollection and an IFDDataObjectCollection. (Typically the sample collection would consist of only one IFDSample object, but we allow for the possibility that, for example, multiple samples have contributed to the linked data object collection.

2.5.1.2 IFDStructureDataAssociation

This class is the key class for connecting a chemical structure with its spectroscopic data, containing two collections, an IFDStructureCollection and an IFDDataObjectCollection. In the simplest case, this would be one structure and its associated spectrum, but it can represent all possible relationships: one-to-one (the structure associated with a spectrum), one-to-many (a set of spectra associated with a give structure), many-to-one (a set of structures describing a mixture associated with a spectrum), and many-to-many (a set of spectra associated with a mixture of compounds).

2.5.1.3 IFDSampleStructureAssociation

This class indicates the connection between (one or more) physical samples and (one or more) chemical structures. It is intended to be used specifically when there is no data object in the collection that links a given sample with a given structure, or when there is no such data *needed* to make that association. For example, the sample is a commercial sample, and the quality of that sample can be ensured. Or the sample's chemical identity is known, but the linking data are not available. or the way the digital entities were ingested into the collection made the logical connection between a "sample" (perhaps the "compound number" of a journal article") and one or more chemical structures (drawn by the author and given that specific compound number).

2.5.2 three-element "analysis" associations

class IFDStructureDataAnalysis extends IFDAssociation
class IFDStructureDataAnalysisCollection extends IFDAssociationCollection

2.5.2.1 IFDStructureDataAnalysis

This class is intended to represent a detailed correlation between chemical structure for a compound and its related experimental or simulated spectroscopic data. It contains three collections, an IFDStructureCollection, an IFDDataObjectCollection, and an IFDAnalysisObjectCollection. For instance, it might correlate specific atoms or groups of atoms of a chemical structure with specific signals in a spectrum or other sort of data object. The analysis object's representations would hold the analysis itself. Some analyses will need to involve specific representations of a structure in order to identify specific atoms in the structure by name or number. (We note that Jmol SMARTS selection syntax could also be used in this respect to describe a specific atom in a structure without regard to any particular numbering system.)

org.iupac.fairdata.derived.IFDSampleDataAssociation
org.iupac.fairdata.derived.IFDSampleStructureAssociation
org.iupac.fairdata.derived.IFDStructureDataAssociation
org.iupac.fairdata.derived.IFDStructureDataAnalysis

inherits fields of IFDAssociation and IFDObject
adds no additional fields

org.iupac.fairdata.derived.IFDSampleDataAssociationCollection
org.iupac.fairdata.derived.IFDSampleStructureAssociationCollection
org.iupac.fairdata.derived.IFDStructureDataAssociationCollection
org.iupac.fairdata.derived.IFDStructureDataAnalysisCollection

inherits fields of IFDAssociationCollection, IFDCollection, and IFDObject
adds no additional fields

3. The IUPAC FAIRSpec extension of the IUPAC FAIRData Metadata Object Model

Note that nothing described above limits IUPAC FAIRData Metadata Object Model to spectroscopy. The IUPAC FAIRData Metadata Object Model is designed to handle any sort of data and their associations to physical samples and chemical structures. Extensions of the model can be accomplished in two ways. First, the objects of the model can be subclassed to add additional components and relationships. Second, the metadata keys and values themselves can be extended. The IUPAC FAIRSpec extension involves both of these types of extensions.

3.1 FAIRSpec Metadata object extensions

3.1.1 FAIRSpecFindingAid and FAIRSpecCollection classes

class FAIRSpecFindingAid extends IFDFindingAid
class FAIRSpecCollection extends IFDCollectionSet

FAIRSpecFindingAid adds a "brand" to an IFDFindingAid by declaring its type to be *org.iupac.fairdata.contrib.FAIRSpecFIndingAid*. This indicates that this particular IFDFindingAid will contain metadata specific to the area of spectroscopy. When serialized, an object of this class becomes the IUPAC FAIRSpec Finding Aid digital object that ultimately distinguishes an IUPAC FAIRSpec Collection from other sorts of digital collections. The serialization of a FAIRSpecFindingAid object might be the target of a persistent identifier such as a DOI (Digital Object Identifier) for the collection (if XML) or interpreted by that DOI's landing page (if JSON). Or, it could be dynamically created in response to a query, allowing standardized programmatic access to the collection. A JSON serialization example is illustrated below and can be found at https://chemapps.stolaf.edu/iupac/site/ifd2/acs.joc.0c00770._IFD_findingaid.json

FAIRSpecCollection, like FAIRSpecFindingAid, brands the collection as utilizing spectroscopic-specific properties, reporting its type as *org.iupac.fairdata.contrib.FAIRSpecCollection*.

org.iupac.fairdata.contrib.fairspec.FAIRSpecFindingAid inherits fields from IFDFindingAid and IFDObject		
field name	**data type**	**description**
citations[ ]	Map<String,?>	a map of named collections

Note that the *contents* guide is for human readability and convenience only. Nothing in this section is essential to the finding aid itself. Some implementations may provide this section as the result of a preview query prior rather than replying initially with a fully elaborated finding aid.

org.iupac.fairdata.contrib.fairspec.FAIRSpecCollection
inherits fields of IFDCollectionSet, IFDCollection, and IFDObject
adds no additional fields

3.1.2 Instrumental technique-specific subclasses

class FAIRSpecCOMPData
class FAIRSpecCOMPDataRepresentation
class FAIRSpecIRData
class FAIRSpecIRDataRepresentation
class FAIRSpecNMRData
class FAIRSpecNMRDataRepresentation
class FAIRSpecMSData
class FAIRSpecMSDataRepresentation
class FAIRSpecHRMSData
class FAIRSpecHRMSDataRepresentation
class FAIRSpecRamanData
class FAIRSpecRamanDataRepresentation
class FAIRSpecUVVISData
class FAIRSpecUVVISDataRepresentation

These technique-specific classes declare an IUPAC FAIRSpec object with additional technique-specific properties but do not add any new fields.

org.iupac.fairdata.contrib.fairspec.dataobject.ir.FAIRSpecCOMPData
org.iupac.fairdata.contrib.fairspec.dataobject.ir.FAIRSpecIRData
org.iupac.fairdata.contrib.fairspec.dataobject.nmr.FAIRSpecNMRData
org.iupac.fairdata.contrib.fairspec.dataobject.ms.FAIRSpecMSData
org.iupac.fairdata.contrib.fairspec.dataobject.hrms.FAIRSpecHRMSData
org.iupac.fairdata.contrib.fairspec.dataobject.raman.FAIRSpecRamanData
org.iupac.fairdata.contrib.fairspec.dataobject.uvvis.FAIRSpecUVVISData

inherits fields of IFDDataObject, and IFDObject
adds no additional fields

org.iupac.fairdata.contrib.fairspec.dataobject.comp.FAIRSpecCOMPDataRepresentation
org.iupac.fairdata.contrib.fairspec.dataobject.ir.FAIRSpecIRDataRepresentation
org.iupac.fairdata.contrib.fairspec.dataobject.nmr.FAIRSpecNMRDataRepresentation
org.iupac.fairdata.contrib.fairspec.dataobject.ms.FAIRSpecMSDataRepresentation
org.iupac.fairdata.contrib.fairspec.dataobject.hrms.FAIRSpecHRMSDataRepresentation
org.iupac.fairdata.contrib.fairspec.dataobject.raman.FAIRSpecRamanDataRepresentation
org.iupac.fairdata.contrib.fairspec.dataobject.uvvis.FAIRSpecUVVISDataRepresentation

inherits fields of IFDDataObjectRepresentation
adds no additional fields

3.1.2 Instrumental technique-specific properties

The IUPAC FAIRSpec extension adds a growing list of metadata key/value pairs specific to particular instrumental techniques. These definitions will evolve; current definitions can be found in the fairspec.properties file at the IUPAC GitHub site. As of this version, for example, NMR properties include:

IFD.property.dataobject.fairspec.nmr.expt_dim
IFD.property.dataobject.fairspec.nmr.expt_freq1
IFD.property.dataobject.fairspec.nmr.expt_freq2
IFD.property.dataobject.fairspec.nmr.expt_freq3
IFD.property.dataobject.fairspec.nmr.expt_nucl1
IFD.property.dataobject.fairspec.nmr.expt_nucl2
IFD.property.dataobject.fairspec.nmr.expt_nucl3
IFD.property.dataobject.fairspec.nmr.expt_pulse_prog
IFD.property.dataobject.fairspec.nmr.expt_solvent
IFD.property.dataobject.fairspec.nmr.expt_temperature_absolute
IFD.property.dataobject.fairspec.nmr.expt_title
IFD.property.dataobject.fairspec.nmr.instr_freq_nominal
IFD.property.dataobject.fairspec.nmr.instr_manufacturer_name
IFD.property.dataobject.fairspec.nmr.instr_probe_type

Appendix A. Recognized IUPAC FAIRData Data Types

Data Type	**Java Equivalent**	**Python Equivalent**	**C++ Equivalent**
Byte	byte	byte	char
Integer	int or long*	int	int
Double	double	float	double
Boolean	boolean	bool	bool
String	String (UTF-8)	string (UTF-8)	std::string
byte	byte	byte	char
array	List or Array	list	array
Map	Map	dict	std::map

*The model does not make a distinction between some implements of data type. In this specification, we will use them synonymously. We will use List specifically when we want to qualify it as a

Appendix B. Full list of IUPAC FAIRData Properties

IFD.property.collectionset.len
IFD.property.collectionset.ref
IFD.property.collectionset.source_repository_uri
IFD.property.collectionset.source_publication_uri
IFD.property.collectionset.source_data_uri
IFD.property.collectionset.source_data_license_uri
IFD.property.collectionset.source_data_license_name

Appendix C. Full list of IUPAC FAIRData MetaData-Related Classes

B.1 core classes (11)

org.iupac.fairdata.core.IFDAssociation
org.iupac.fairdata.core.IFDAssociationCollection
org.iupac.fairdata.core.IFDCollection
org.iupac.fairdata.core.IFDCollectionSet
org.iupac.fairdata.core.IFDFindingAid
org.iupac.fairdata.core.IFDObject
org.iupac.fairdata.core.IFDProperty
org.iupac.fairdata.core.IFDReference
org.iupac.fairdata.core.IFDRepresentableObject
org.iupac.fairdata.core.IFDRepresentation
org.iupac.fairdata.core.IFDResource

B.2 primary subclasses of IFDRepresentableObject, and related classes (12)

org.iupac.fairdata.analysis.IFDAnalysisObject
org.iupac.fairdata.analysis.IFDAnalysisObjectCollection
org.iupac.fairdata.analysis.IFDAnalysisObjectRepresentation

org.iupac.fairdata.dataobject.IFDDataObject

org.iupac.fairdata.dataobject.IFDDataObjectCollection
org.iupac.fairdata.dataobject.IFDDataObjectRepresentation

org.iupac.fairdata.sample.IFDSample
org.iupac.fairdata.sample.IFDSampleCollection
org.iupac.fairdata.sample.IFDSampleRepresentation

org.iupac.fairdata.structure.IFDStructure
org.iupac.fairdata.structure.IFDStructureCollection
org.iupac.fairdata.structure.IFDStructureRepresentation

B.3 derived associations and related classes (8)

org.iupac.fairdata.derived.IFDSampleDataAssociation
org.iupac.fairdata.derived.IFDSampleDataAssociationCollection

org.iupac.fairdata.derived.IFDSampleStructureAssociation
org.iupac.fairdata.derived.IFDSampleStructureAssociationCollection

org.iupac.fairdata.derived.IFDStructureDataAnalysis
org.iupac.fairdata.derived.IFDStructureDataAnalysisCollection

org.iupac.fairdata.derived.IFDStructureDataAssociation
org.iupac.fairdata.derived.IFDStructureDataAssociationCollection