
Preliminary reference guide to data exchange with dxf2

Bob Jolliffe

Table of Contents

Introduction and rationale	1
Looking at some sample messages	2
A routinely reported dataset from a single facility	2
A routinely reported dataset across multiple facilities	3
Dimensional data	3
Missing pieces	4

Introduction and rationale

This document describes the current state of development of the data exchange format for DHIS2, referred to as dxf2. Neither the format nor its implementation is complete, but certain functionalities are already available within DHIS2, so this document serves to describe what is there, as well as provide a roadmap and basis of discussion for the future. Whereas some pointers are given regarding metadata exchange, the current format, as described here, deals specifically with the importing of data (ie data values) into DHIS2.

The evolution of dxf2 from the earlier dxf1 to what it is currently, is driven by the following motivating factors:

1. The desire to separate the importing of data from the importing of metadata. In dxf1 a data message is prefixed by a "header" which contains the complete metadata description of all the orgunits, periods, dataelements and dimensional structures referred to in the data values. Metadata import and id resolution takes place as a required step before importing the data values. Whereas this approach can have some value, particularly when bootstrapping new databases, it is not suitable as a general means of routinely importing data because,
 - for small messages, for example dataset reporting via mobile phone, the metadata overhead is excessive and largely redundant;
 - for third party systems, the generation of such a complex header is difficult;
 - reimporting metadata on a routine basis, including from untrusted sources, poses a hazard to the integrity of the metadata within production systems.
2. To reinforce better practice of identification of metadata within DHIS2 and amongst co-operating systems. Currently identifiable elements within DHIS2 (eg. dataelements, orgunits etc) are uniquely distinguishable either by database primary key identifier or by the *name* field of the element. Neither of these two represent good practice and nor do they reflect the reality of the world around DHIS2. Different databases will use different values of primary key and different systems will name things differently¹. Whereas the exorcising of the use of names as unique identifier in DHIS2 is a more complex project, the aim here is to not propagate or otherwise expose that practice in our interaction with other systems.
3. To similarly contain the idiosyncrasy of DHIS2 representation of dimensions of data within the system and not expose it to external systems.
4. To handle the transmission of time periods in a way which does not require the synchronisation of coded periods between systems.

Guided by the rationale above, the new dxf2 defines a `datavalueset` element which is designed to contain a sequence of `datavalues`, without the requirement of a metadata header. Structural metadata items can be identified by either one of *uuid*, *code* or *internal id*. Despite efforts over the past year to imagine into existence the one true globally unique identifier, the world is more complex and different forms of identifier may be suited to different situations². Periods are represented as ISO8601 coded strings according to the `toString()` methods within the period API. Examples include "201109", "2011W32", "2011Q1", "2011" for monthly, weekly, quarterly and yearly period types respectively.

Looking at some sample messages

Describing a format by means of examples is necessarily informative rather than normative. For the purpose of this document that is sufficient. A normative document and authoritative schema will emerge later. The first two of these examples can be imported through the `dhis2 import/export` module, provided of course that the codes or uuids are defined. The current schemas, as implemented within DHIS2, is available at

A routinely reported dataset from a single facility

Use cases: importing data from a patient based system in a hospital, importing data from community health workers via mobile.

```
<dxf xmlns="http://dhis2.org/schema/dxf/2.0"
  minorversion="1.0" date="20110913">
  <dataValueSets>
    <dataValueSet idScheme="uuid"
      dataSet="16B2299E-ECD6-46CF-A61F-817D350C180D"
      period="2011W5"
      orgUnit="9C1B1B5E-3D65-48F2-8D1D-D36C60DD7344">
      <dataValue dataElement="56B2299E-ECD6-46CF-A61F-817D350C180D" value="11" />
      <dataValue dataElement="56B2299E-ECD6-46CF-A61F-817D350C180E" value="11" />
      <dataValue dataElement="56B2299E-ECD6-46CF-A61F-817D350C180F" value="11" />
      <dataValue dataElement="56B2299E-ECD6-46CF-A61F-817D350C180G" value="11" />
    </dataValueSet>
  </dataValueSets>
</dxf>
```

Some notes on the above

- The enclosing `dxf` root element is not a strict requirement. Particular consumers, eg a custom webservice for reading mobile messages, might choose to implement just the `dataValueSet`.
- The `minorversion` attribute is a useful way of indicating minor changes to consumers without having to trigger changes in namespace.
- The date the message was produced is perhaps better implemented as a timestamp. Please vote.
- The `idScheme` determines the type of identification used. In this case that is `uuid`. This attribute is an indication to the consumer to match against the `uuid` field of identifiable objects within DHIS2.
- The `dataset` is an optional attribute which is strongly encouraged. In the absence of a `dataset` attribute the potential list of valid `orgunits` and `dataelements` is unconstrained. Providing a `dataset` attribute

²It is my view that *code* will in most cases be the preferred identifier, but the other two remain as options. It is also my view that we will eventually expand the notion of *code* to allow different codes for different contexts, but for the moment, this requirement has not emerged concretely.

allows consumers to validate that only appropriate orgunits and dataelements are reported. There may be instances where this is not desirable, for example a third party system reporting dataelements from across DHIS2 datasets. Another benefit of the dataset attribute is that it can be used for checking against the credentials of the user importing this set.

- The period attribute refers to a string representation of the period as discussed above. Note that it is defined at the dataValueSet level, so it is effectively inherited for each of the dataValues.
- The orgUnit attribute is similarly defined at the dataValueSet level. There is a class of use cases illustrated in the next example where it may not be appropriate to define this attribute here.
- A dataValue consists of at least two attributes, a dataElement and a value. The schema is however permissive in the sense that there may be *any additional attributes*. This is to allow for the specification of orgunits at the dataValue level as illustrated below, as well as to cater for the inclusion of arbitrary dimension attributes.

A routinely reported dataset across multiple facilities

Use cases: importing data from another aggregate-type system, such as a human resource, logistical or meteorological type systems which will have data from across a geographical spread.

```
<dxf xmlns="http://dhis2.org/schema/dxf/2.0"
      minorversion="1.0" date="20110913">
  <dataValueSets>
    <dataValueSet idScheme="code"
                  dataSet="IHRIS_HUMAN_RESOURCE"
                  period="2011Q2" >

      <dataValue dataElement="145" orgUnit="14356" value="11" />
      <dataValue dataElement="145" orgUnit="14356" value="12" />
      <dataValue dataElement="145" orgUnit="14356" value="11" />
      <dataValue dataElement="145" orgUnit="14356" value="9" />
    </dataValueSet>
  </dataValueSets>
</dxf>
```

Some notes on the above

- This example illustrates the use of code as identifier scheme. Identifiable objects will be matched against the code field within DHIS2. Note that the code field is an arbitrary string. Note also that we do not necessarily have autonomy on the assignment of codes. For example, the above message is based on a sample from Kenya, where the code is assigned by a 3rd party authority which maintains a master facility list.
- In this case, the orgUnit code is not assigned at the dataValueSet level.

Dimensional data

Use cases: importing data which is disaggregated. Note that this feature is not yet implemented as it requires the fuller implementation of concepts envisaged in the next version of DHIS2.

```
<dxf xmlns="http://dhis2.org/schema/dxf/2.0"
      minorversion="1.0" date="20110913">
  <dataValueSets>
```

```
<dataValueSet idScheme="code"
              dataSet="IHRIS_HUMAN_RESOURCE"
              period="2011Q2" >

  <dataValue dataElement="145" orgUnit="14356" value="11" sex='M' age='under5
  <dataValue dataElement="145" orgUnit="14356" value="12" sex='F' age='under5
  <dataValue dataElement="145" orgUnit="14356" value="11" sex='M' age='5andOv
  <dataValue dataElement="145" orgUnit="14356" value="9" sex='F' age='5andOve
</dataValueSet>
</dataValueSets>
</dxf>
```

Some notes on the above

- The datavalue is augmented by two additional parameters, age and sex. These are equivalent to a single categorycombo attribute. It is conceivable that a categorycombo attribute could also be used for tightly coupled systems, but its not encouraged.
- The names of the attributes will be matched against concept names in DHIS2.
- The "validity" of the attribute set is determined by the value of the dataElement code ie. It cannot be strictly validated with declarative schema.
- Given that we are using idScheme code, the values of the parameters are matched using codes. These codes could be numeric, but can equally be more human understandable.

Missing pieces

The data format as described above, is suited to represent the variety of data exchange use cases we have encountered, once the dimensional values are implemented. Two critical remaining pieces need to be implemented:

Response message	Dxf1 provides no means to return a response to a datavalue message. The only feedback is available via the transient status message within the user interface and through the logs. Dxf2 needs to define a response message which can be interpreted by 3rd part systems and/or rendered through the web user interface, indicating the result of the import, validation etc.
Metadata messages	In addition to the xml representation of the entire metadata map, dxf2 needs to provide a simple means of producing and consuming codelists for the purpose of establishing the basis for data exchange.