

# User Guide - November 2010 Schema release

## User Guide - November 2010 Schema release (1.1)

### Introduction to the V/FRBR XML Schemas

The V/FRBR: Variations as a Testbed for the FRBR Conceptual Model project aims to implement the FRBR conceptual model in a production system, for the purpose of better understanding how FRBR might be used to build next-generation catalogs. One key part of this project is to understand what it means for real *production* data to "conform" to FRBR. To this end, the V/FRBR project team has developed a set of XML Schemas that we hope will eventually grow into a best practice for the representation of FRBRized bibliographic data in XML.

The full package of V/FRBR XML Schemas is made up of 132 W3C XML Schema documents divided into 3 "levels" as discussed later in this User Guide, along with sample XML instance documents and supporting documentation. This set V/FRBR XML Schemas are available in two forms:

- As a set of files posted on a publicly-accessible web server, at <http://vfrbr.info/schemas/>
- As a set of files wrapped in a downloadable package, at <http://vfrbr.info/schemas/schemas.zip>

The online files are intended for quick reference, and as authoritative, up to date versions that may be referenced by live applications. Each Schema in this set references others when necessary through absolute URLs to these web-accessible versions. The downloadable files are intended for study and use in applications where local copies of Schema documents are desirable. Schemas in this set reference others through relative URLs, allowing the data in this downloaded package to be re-used in local applications.

The 1.1 Schema release is a refinement of our March 2010 1.0 release, addressing several issues we found with the earlier Schemas as we worked with them. As a project, we decided early on that an XML data representation format should be our highest priority, as we believed XML was a better fit for the current library technology environment than RDF. We view this XML format as merely one external representation of the data (not as our internal system representation). In fall 2010, we began work on an RDF representation of this data as well, and expect to release data in this form and specifications for our RDF modeling in the next few months. The 1.1 XML Schemas reflect some of the same limitations present in the 1.0 versions; for example, at this time we implement only a small proportion of the model presented in the FRAD report; we incorporate the Family entity and additional attributes FRAD defines on the original FRBR Group 1, 2, and 3 entities, but do not at this point model the other additional entities defined in the FRAD report such as Name and Controlled Access Point. We will continue to update these Schemas through the life of the project.

### Design goals

The XML Schemas designed for the Variations/FRBR project should support our project priorities of providing a replicable model for other FRBRized catalogs and for implementing FRBR in the production Variations system. As such, here are our primary design goals for these Schemas:

1. Provide a re-usable model for representing data in XML that conforms to the FRBR model for other systems implementing FRBR.
2. Represent in an XML format:
  - a. FRBR Group 1 entities and their attributes,
  - b. FRBR Group 2 entities and their attributes,
  - c. relationships defined in the FRBR report,
  - d. any features from FRAD we determine provide a good balance between our local project needs and the role of these Schemas as a model for others,
  - e. and *possibly* FRBR Group 3 entities and their attributes.
3. Allow re-usability of FRBR/FRAD entities, FRBR/FRAD attributes, and FRBR/FRAD relationships by other applications, in whole or in part.
4. Provide a best practice packaging format for grouping together XML representations of FRBR/FRAD entities, FRBR/FRAD attributes, and FRBR/FRAD relationships for transport between systems and in response to Web Services requests.
5. Promote human understandability of XML instance documents conforming to our XML Schemas by:
  - a. Using XML element and XML attribute names that match terminology in the FRBR and FRAD reports, and
  - b. Creating a general structure for XML instance documents that reflects the structure and intellectual divisions in the FRBR and FRAD reports.
6. Provide options for encoding data according to:
  - a. an extremely strict interpretation of FRBR,
  - b. an extension of FRBR for general use with what we judge to be the minimum that makes FRBR data "useful," and
  - c. an extension and contraction of FRBR optimized for use with musical scores and recordings.
7. Separate and connect together the definition of FRBR/FRAD entities, FRBR/FRAD attributes, and FRBR/FRAD relationships in a manner that effectively promotes and defines a best practice for other implementers both extending and restricting FRBR while re-using the XML structures our Schemas provide.

### Scope

For the V/FRBR project, we expect to represent data in XML instance documents that conform to our XML Schemas, within the following scope:

1. Musical works and their representations as scores and recordings, some of which might have accompanying textual material,
2. Non-musical works related to musical works that potentially provide valuable discovery services for end users, such as textual works that have been set to music either strictly or liberally,
3. Only create XML instances for entities that are represented in the V/FRBR system,
4. Only use as the source and target for relationships entities that are represented in the V/FRBR system,
5. As responses to external Web Services requests, and
6. For exchange of data between Variations and V/FRBR tools to the lowest level reasonably expected to need future interoperability with other tools built by IU and others.

## Schema Structure

### Levels

In order to accommodate our competing design goals of providing a model for FRBRized data in library catalogs of the future and for adequate representation of music metadata in a FRBR structure for local use, we have developed a 3-tiered XML Schema structure. The first and simplest level is "frbr" which represents the entities and attributes described in the FRBR and parts of the FRAD reports faithfully, with a minimum of additional features added. The second level is "efrbr" (short for "extended frbr") which we expect will be of the most use to other FRBR implementers, as it adds features intended to make the FRBR model useful in practice for actual data storage and use. The third level is "vfrbr" (short for "Variations/FRBR") which is intended for the representation of V/FRBR project-specific data, including extensions to FRBR we believe are essential for the representation of musical materials.

#### frbr

The frbr level of XML Schemas simply defines XML elements for each FRBR entity and subelements for each FRBR attribute. XML element names are taken directly from the FRBR report, with spaces removed, and converted into camel case. All FRBR attributes are defined as XML elements with a string data type. No element is required and all are repeatable. The only addition to the FRBR report that XML Schemas at this level make is an @identifier attribute for each XML element representing a FRBR entity. This addition allows relationships to be created between entities. 36 XML Schema files make up the frbr level representation.

#### efrbr

The efrbr level of XML Schemas adds significant value to the data structure implied by the conceptual model presented in the FRBR report. A new <note> is added to each FRBR entity to allow the recording of important data which has no other natural place in the FRBR model. In addition, we have added XML attributes to many XML elements representing FRBR attributes to allow for the further refining of their meaning and the disambiguation of multiple values from one another. These attributes have been applied selectively to only the XML elements to which we believe they apply. These attributes are:

- @type: used in many locations as an element refinement
- @jurisdiction: appears on XML elements representing place names, and used to indicate the jurisdictional level of the place (e.g., city, state, country)
- @function: appears on some date elements to allow an indication of what the date applies to: birth, death, creation, dissolution, etc.
- @timeframe: used to flag whether data such as a name or location is "current" or "former"
- @offset: records the number of non-filing characters present in an element's content
- @quantity: used in cases such as medium of performance (instrumentation) when the element content is controlled but it is also necessary to provide an indication of quantity
- @normal: records a normalized form of free-form data in element content; used frequently for dates and controlled lists which exist in both coded and textual versions
- @vocabulary: used to indicate the controlled vocabulary from which the element content has been selected
- @availability: flags whether data should be visible to end-users or only internal users

In addition, the efrbr level groups together three publication elements (place of publication/distribution, publisher/distributor, date of publication/distribution) into a publicationDistribution group, to allow for connecting different publication/distribution dates to different publishers or distributors. 48 XML Schema files make up the efrbr level representation.

#### vfrbr

The vfrbr level of XML Schemas is intended to record the data necessary for the operation of the Variations/FRBR discovery system. It simultaneously restricts data from efrbr (removing support for FRBR attributes not applicable to the description of musical materials) and adds new data through importing XML Schemas that define extensions to FRBR for musical materials. 48 XML Schema files make up the vfrbr level representation.

Like efrbr, the vfrbr level includes a <note> element, adds XML attributes for additional functionality, and groups together the publication-related elements.

## Packaging format

Each of the three levels of XML Schemas (frbr, efrbr, and vfrbr) defines a record packaging format. The packaging format is identical at each level, differing only in its target namespace. For each level, this namespace defines a top-level <recordSet> element, which in turn has child elements <entities> and <relationships> in the same namespace that group entity and relationship data, respectively.

We have provided sample packages showing what data at each of the three levels (frbr.xml, efrbr.xml, and vfrbr.xml) might look like. The wrapping package can be used to represent any arbitrary group of FRBR entities and the relationships between them. The sample frbr.xml, efrbr.xml, and vfrbr.xml packages show a common use case, a Manifestation held by a library, a digitized version of that Manifestation, all other Group 1, 2, and 3 entities connected to that Manifestation, and explicitly stated relationships between them. The packaging structure is designed to be used for other groupings of FRBR data as well; for example, all Works that meet query criteria, all Expressions of a given Work, or all Group 2 entities with relationships to a given Work or Expression.

## FRBR entities and their namespaces

Within the <entities> element for each level, the primary frbr/efrbr/vfrbr Schemas use the xsd:import method to bring in separate Schemas for each FRBR Group 1, 2, and 3 entity. Each of these entity-level Schemas (the "entity" links on the schema listing directory page) defines its own target namespace, to distinguish between FRBR attributes that have the same name on more than one entity (for example, both Work and Expression have a medium of performance attribute). The primary entity element in each of these schemas allows an @identifier attribute for a unique identifier for that entity, which can then be referenced later in the <relationships> section of an instance document. When FRAD defines new attributes for a FRBR entity, these additions are defined in their own Schema, in a FRAD namespace (the "FRAD" links on the schema listing directory page), then are imported into the FRBR entity-level Schema. The additional FRAD Group 2 entity Family is similarly defined with an entity-specific target namespace showing its FRAD (rather than FRBR) origins, then imported into the main Schema.

Each entity-level Schema uses the xsd:include method to bring in a separate Schema (in the same namespace) that defines the FRBR attributes (XML subelements) for that entity (the "attributes" links on the Schema listing directory page). These are defined in a separate Schema (the "attributes" links on the Schema listing directory page) to ensure their definitions are global, which promotes re-use of these FRBR attributes. They however do not need to be defined in their own namespace, therefore each uses the same namespace as the entity to which the attributes apply.

At the efrbr and vfrbr levels, each of the Schemas defining the FRBR attributes for a given entity uses the xsd:include method to bring in the same Schema (the "Common Entity Types" link on the Schema listing directory page) that defines a set of common data types. These types are used to add selected XML attributes to the XML elements representing FRBR attributes. Because the data types are the same for each entity, the Schema defining them does not contain a target namespace to allow these types can be re-used by many different Schemas.

Each of the entity-level Schemas at the efrbr and vfrbr levels uses the xsd:include method to bring in a Schema defining some extensions to the base FRBR attributes. Each of these adds a <note> element in the entity's namespace, as well as an <extensions> element that is designed to allow locally-namespaced additions to the base FRBR model. Both are defined in the Schemas available from the "extensions" links on the schema listing directory page. At the vfrbr level, this <extensions> element is defined to contain the additions to FRBR our project has identified as necessary.

A diagram illustrating the various Schema imports and includes used may be found later in this Guide. Also included in the downloadable package with the set of XML Schemas is an "element definitions" spreadsheet showing at a glance the data elements available at the frbr, efrbr, and vfrbr levels.

## FRBR relationships and their namespaces

The method for recording relationships is the primary difference between the V/FRBR 1.0 XML Schemas and the 1.1 versions. Within the <relationships> element for each level, the XML packaging structure is further divided into segments for structure relations, responsible relations, subject relations, and other relations. These divisions are based on the way the original FRBR report groups and describes FRBR relationships. Structure relations are those between the Group 1 entities (Work, Expression, Manifestation, and Item): Works are *realized through* Expressions, Expressions are *embodied in* Manifestations, and Manifestations are *exemplified by* Items. Responsible relations are those between Group 2 entities and Group 1 entities: Works are *created by* Group 2 entities, Expressions are *realized by* Group 2 entities, Manifestations are *produced by* Group 2 entities, and Items are *owned by* Group 2 entities. Subject relations are those between Works and any Group 1, 2, or 3 entity that serves as a subject of that work: here a work *has subject* any Group 1, 2, or 3 entity. Other relations include additional relationships between specific types of entities: Work to Work, Expression to Expression, Expression to Work, Manifestation to Manifestation, Manifestation to Item, and Item to Item.

The relationship type segments further include XML elements for specific FRBR (and now in the 1.1 versions, FRAD) relationships, such as realizedThrough or createdBy. Each of these XML elements for specific relationships has the same definition. They are empty, with all meaning provided via their attributes. There are attributes for the type of entity for the source of the relationship, the URI of the source entity, the type of entity for the target of the relationship, and the URI of the target entity. Many of the FRBR relationships have only certain types of entities as valid on one side of the other of a relationship (for example, the hasSubject relationship can only have a Work as its source and can have any of the Group 1, 2, or 3 entities as its target). To accommodate this, the specific relationship XML elements are defined to restrict the value of the attributes for source and target entities to reasonable values. The hasPart relationship is an exception to this general approach, however, as many types of FRBR entities have part/whole relationships to other entities. The V/FRBR 1.1 XML Schemas take a generic approach to the hasPart relationship, allowing any Group 1 entity to be both the source and target, instead of restricting data to valid combinations from the FRBR report.

The specific responsible and other relationship elements also include additional attributes. For relationships in the responsible group, a role attribute is available to allow more detailed specification of the contribution of a Group 2 entity to a Group 1 entity, for example, if a Person is the composer or the lyricist for a given Work. This attribute takes a string value; however, local control would likely be applied to this data, perhaps using MARC relator codes. For relationships in the other group, a category attribute is available to allow for the further specification of the type of relationship as found in the FRBR report, for example, if a Work is an autonomous or referential supplement to another Work. Values for the category attribute are provided based on charts in the FRBR report, where applicable.

All of the relationships across the categories are represented in these XML formats as in a single direction only, however, each is assumed as defined in the FRBR report to be reciprocal. The language of the relationship element indicates to a human reader the desired direction.

One primary XML Schema exists at each level for each of the four categories of FRBR relationships. These primary relationship Schemas (the

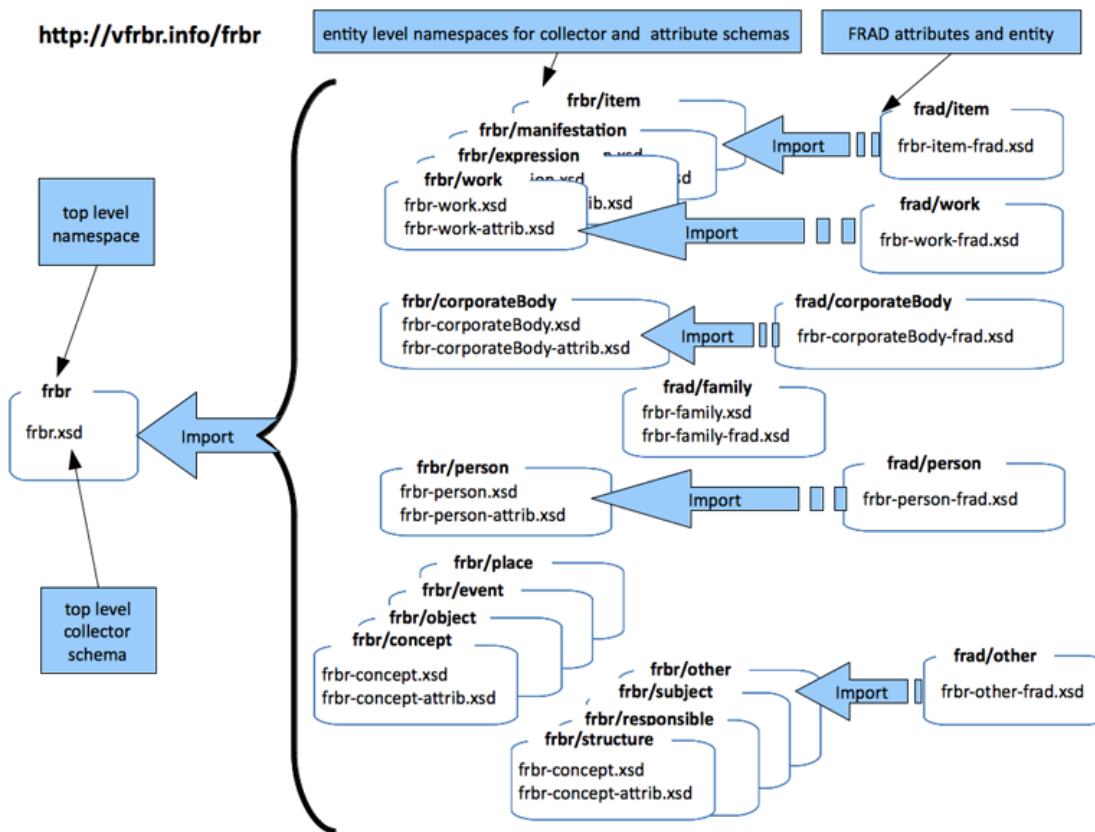
"relation" links on the schema listing directory page) define these top-level relationship categories. Each of these Schemas uses the `xsd:include` method to bring in a support Schema defining the XML attributes of the specific relationship element, including restrictions on what entity types are valid for the source and target of each relationship (the "attributes" links on the Schema listing directory page). Each of the four top-level categories of FRBR relationships has its own namespace.

A diagram illustrating the various Schema imports and includes used may be found later in this Guide.

## Namespace policies

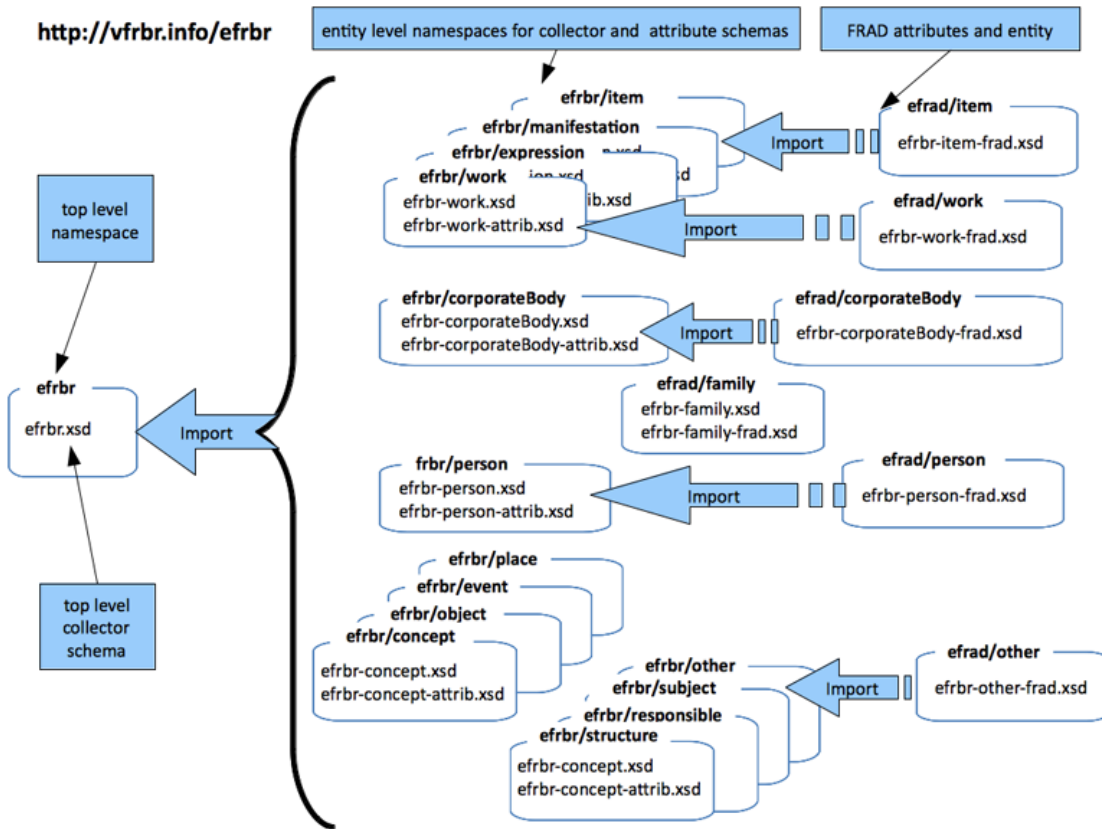
The `frbr`, `efrbr`, and `vfrbr` levels of schemas each are defined with their own level-specific target namespaces, with the primary form of <http://vfrbr.info/frbr>, <http://vfrbr.info/efrbr>, and <http://vfrbr.info/vfrbr>, respectively. The significant difference in data type for the FRBR attributes between the `frbr` and `efrbr` levels makes it necessary to completely redefine all data at each level rather than attempting to build one level on the next. Specific FRBR entities then build upon the base target namespaces when necessary, for example <http://vfrbr.info/frbr/work>. The following diagrams illustrate how each level brings together the many Schemas that make up its definition.

### frbr level namespace structure



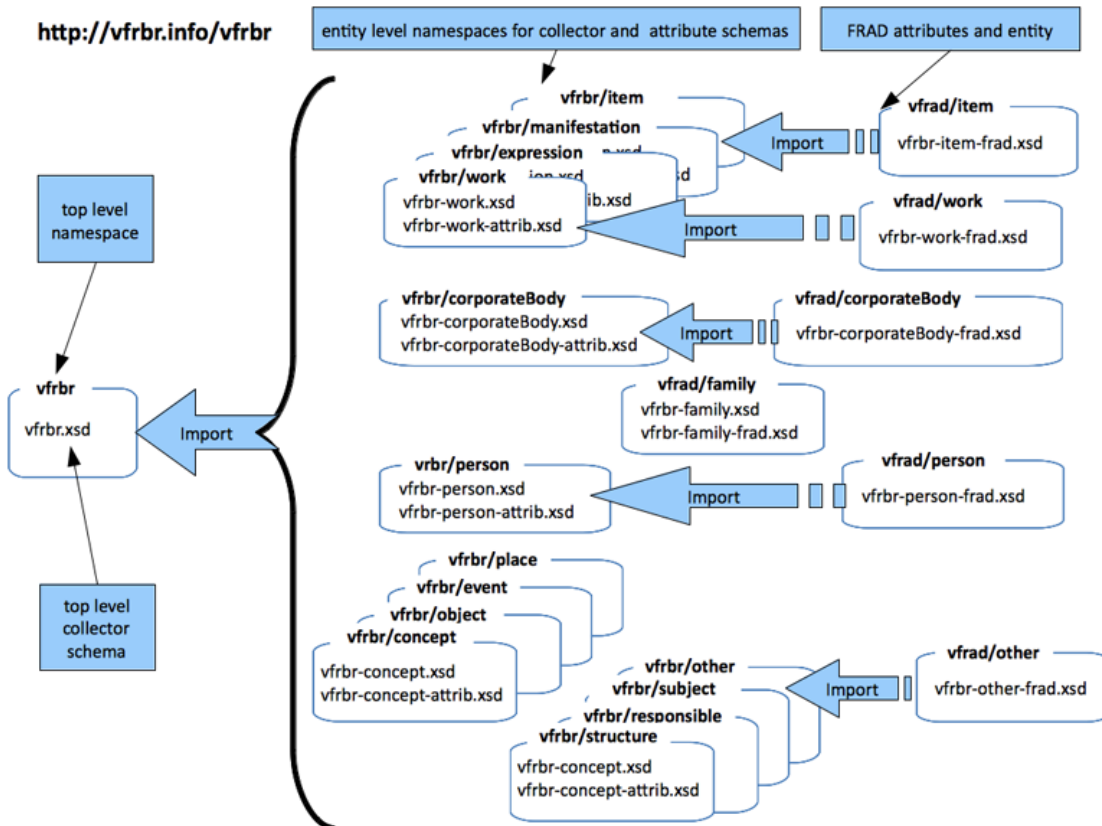
### efrbr level namespace structure

http://vfrbr.info/efrbr



### vfrbr level namespace structure

http://vfrbr.info/vfrbr



## We're seeking feedback!

While this 1.1 XML Schema release represents an update, there are still areas in which these Schemas we could improve, including fuller support for FRAD and the introduction of administrative, record management data. The V/FRBR team welcomes feedback on any aspect of these XML Schemas at [vfrbr@dlib.indiana.edu](mailto:vfrbr@dlib.indiana.edu). However, we are particularly interested in feedback on the following design decisions:

- The general approach of defining features we believe might be re-usable in other contexts at the Schema level, then import or include these Schemas into others to build up the full structure.
- Our many-namespace approach to the Schemas.
- The three-level approach to allowing pure frbr in some contexts, the more useful general efrbr, and the domain-specific vfrbr.
- The division of relationships into four separate categories based on the structure of the relationships discussion in the FRBR report.

Several members of our development team will receive email sent to [vfrbr@dlib.indiana.edu](mailto:vfrbr@dlib.indiana.edu). We look forward to hearing from you.