Semantic Representation

Version: 3.4
Last updated: 13 June 2017
HTML version: http://vocab.getty.edu/doc (for linking)
PDF version: http://vocab.getty.edu/doc/gvp-lod.pdf (for printing)
Sample queries: http://vocab.getty.edu/doc/queries
Queries UI: http://vocab.getty.edu/queries
Initial version: Vladimir Alexiev, Joan Cobb, Gregg Garcia, Patricia Harpring.
Updates: Vladimir Alexiev, Joan Cobb

Table of Contents
1 Introduction .................................................................................................................. 4
  1.1 The Getty Vocabularies and LOD ........................................................................... 4
    1.1.1 About the AAT ............................................................................................... 4
    1.1.2 About the TGN ............................................................................................. 5
    1.1.3 About the ULAN ......................................................................................... 5
  1.2 Revisions, Review, Feedback .................................................................................. 6
    1.2.1 Revisions ...................................................................................................... 6
      1.2.1.1 Version 1.0 .............................................................................................. 6
      1.2.1.2 Version 1.1 ............................................................................................. 6
      1.2.1.3 Version 1.2 ............................................................................................. 6
      1.2.1.4 Version 1.3 ............................................................................................. 6
      1.2.1.5 Version 2.0 ............................................................................................. 6
      1.2.1.6 Version 2.1 ............................................................................................. 7
      1.2.1.7 Version 3.0 ............................................................................................. 7
      1.2.1.8 Version 3.1 ............................................................................................. 8
      1.2.1.9 Version 3.2 ............................................................................................. 8
      1.2.1.10 Version 3.3 ........................................................................................... 9
      1.2.1.11 Version 3.4 ........................................................................................... 9
      1.2.1.12 Future Versions .................................................................................... 9
    1.2.2 External Review Process .................................................................................. 10
    1.2.3 Providing Feedback ...................................................................................... 10
    1.2.4 Disclaimer .................................................................................................... 10
  1.3 Abbreviations ....................................................................................................... 11
  1.4 RDF Turtle .......................................................................................................... 12
  1.5 Prefixes ................................................................................................................ 12
    1.5.1 External Prefixes .......................................................................................... 13
    1.5.2 Descriptive Prefixes .................................................................................... 14
  1.6 GVP URLs and Prefixes ...................................................................................... 14
    1.6.1 Common GVP URLs .................................................................................. 15
    1.6.2 AAT URLs ................................................................................................ 15
    1.6.3 TGN URLs ................................................................................................ 16
    1.6.4 ULAN URLs ............................................................................................... 16
    1.6.5 Using GVP URLs ....................................................................................... 17
    1.6.6 Named Graphs ............................................................................................ 17
  1.7 Semantic Resolution ............................................................................................ 18
    1.7.1 Semantic Formats ......................................................................................... 18
    1.7.2 Example URLs ............................................................................................. 18
1.8 External Ontologies

1.8.1 DC and DCT

1.8.2 SKOS and SKOS-XL

1.8.3 ISO 25964

1.8.4 BIBO

1.8.5 FOAF

1.8.6 PROV

1.8.6.1 dct:modified

1.8.6.2 dct:creator+dct:created

1.8.7 Geographic Ontologies

1.8.7.1 W3C WGS Geo Ontology

1.8.7.2 Schema.org Geographic Features

1.8.8 Agent Ontologies

1.8.8.1 Bio Ontology

1.8.8.2 Schema.org Agent Features

1.9 GVP Ontology

2 Semantic Representation

2.1 Semantic Overview

2.2 Subject

2.2.1 Subject Types

2.3 Subject Hierarchy

2.3.1 Standard Hierarchical Relations

2.3.2 GVP Hierarchical Relations

2.3.3 Hierarchy Structure

2.3.4 Top Concepts

2.4 Estimated Dates

2.5 Sort Order

2.5.1 Sorting with Thesaurus Array

2.5.1.1 skos:member Structure

2.5.1.2 skos:memberList Structure

2.5.1.3 Full Representation

2.6 Associative Relationships

2.6.1 Relationships Table

2.6.2 Relationship Representation

2.7 Obsolete Subject

2.8 Language

2.8.1 IANA Language Tags

2.8.2 GVP Language Tags

2.8.3 Language Tag Case

2.8.4 Language Tags and Sources

2.8.5 Language Dual URLs

2.9 Term

2.9.1 Term Characteristics

2.9.2 Importance of the Vernacular Flag

2.10 Scope Note

2.11 Identifiers

2.12 Notations

2.13 Source

2.13.1 Local Sources

2.14 Contributor

2.15 Historic Information

2.15.1 Applying to Terms

2.15.2 Applying to Relations and Types
1 Introduction

This document explains the representation of the Getty Vocabularies in semantic format, using RDF and appropriate ontologies.

It covers the Art and Architecture Thesaurus (AAT®), the Thesaurus of Geographic Names (TGN)® and the Union List of Artist Names (ULAN)®. We have included them in the same document because all three share the same basic semantic representation. The documentation will be expanded to include the forth vocabulary, Cultural Objects Name Authority (CONA)®, when it becomes available.

Previously, this document included a section dedicated to ‘Sample Queries’ but, due to the growing size of the samples we are providing, they have been moved to their own document: Sample Queries.

The document is published in HTML format, with appropriately and permanently named anchors for each section that can be shared in discussions. It is also published in PDF appropriate for printing.

1.1 The Getty Vocabularies and LOD

The Getty Vocabularies were first built to help people categorize, describe, and index cultural heritage objects and information. The Getty Vocabularies are compliant with international standards and grow through contribution.

Now we have the technology to transform these human-defined relationships into machine-readable data sets and embed them into the evolving semantic web. By publishing the Getty Vocabularies as Linked Data in an open environment for anyone to freely use, we are sharing with the world the results of over thirty years of research and scholarship.

1.1.1 About the AAT

AAT is a structured, multilingual vocabulary including terms, descriptions, and other information for generic concepts related to art, architecture, other cultural heritage, and conservation. For decades now, the AAT has been used as a primary reference by museums, art libraries, archives, visual resource catalogers, conservation specialists, archaeological projects, bibliographic projects, researchers, and information specialists who are dealing with the needs of these users.

Terms for any concept may include the plural form of the term, singular form, natural order, inverted order, spelling variants, scientific and common forms, various forms of speech, and synonyms that have various etymological roots. Among these terms, one is flagged as the term (or descriptor) preferred by the Getty Vocabulary Program. There may be multiple descriptors reflecting usage in multiple languages. Preferences for individual contributors may differ and are noted.

The AAT is a thesaurus in compliance with ISO and NISO standards.

The focus of each AAT record is a concept. Linked to each concept are terms, related concepts, its position in the hierarchy, sources for the data, and notes. The conceptual framework of facets and hierarchies in the AAT is designed to allow a general classification scheme for art, architecture and conservation. The framework is not subject-specific; for example, there is no defined portion of the AAT that is specific only for Renaissance painting. The terms to describe Renaissance paintings will be found in many locations in the AAT hierarchies. There may be multiple broader contexts, making AAT polyhierarchical.

In addition the AAT has equivalence and associative relationships. The temporal coverage of the AAT ranges from Antiquity to the present and the scope is global.
Currently the AAT includes info about 42k Subjects, 300k Terms, 84k Scope Notes, 40k Sources and 165 Contributors (see queries Counting and Descriptive Info). See more information about the history, purpose and scope of the AAT.

### 1.1.2 About the TGN

TGN is a structured vocabulary containing names and other information about places. Names for a place may be include names in the vernacular language, English, other languages, historical names and in natural and inverted order. Among these names, one is flagged as being preferred by the Vocabulary Program. Preferences for individual contributors may differ and are noted.

TGN is a thesaurus, compliant with ISO and NISO standards for thesaurus construction; it contains hierarchical, equivalence, and associative relationships. Currently there are two facets, World and Extraterrestrial Places. Under the World, places are arranged in hierarchies generally representing the current political and physical world, although some historical nations and empires are also included. There may be multiple broader contexts making the TGN polyhierarchical.

The focus of each TGN record is a place. Linked to each place are names, the place’s position in the hierarchy, other relationships, geographic coordinates, notes, sources for the data, and place types, which are terms from the AAT describing the roles of the place over time (e.g., inhabited place and state capital). The temporal coverage of the TGN ranges from prehistory to the present and the scope is global. See more information about the history, purpose and scope of the TGN.

Note that TGN is not a GIS (Geographic Information System): it is a thesaurus. While many records in TGN include coordinates, these coordinates are approximate and are intended for reference (“finding purposes”) only. Geographic coordinates in TGN typically represent a single point, corresponding to a point in or near the center of the inhabited place, political entity, or physical feature. For linear features such as rivers, the point represents the source of the feature. Some areas and regions (e.g. the Great Lakes region) have bounding box coordinates.

Currently the TGN includes info about 1.26M places (Subjects), 1.85 names (Terms), 22k Scope Notes, 3.1k Sources, 166 Contributors, 1.33M Place Type instances, 722 distinct Place Types

### 1.1.3 About the ULAN

The ULAN is a structured vocabulary that includes proper names or anonymous appellations (e.g., Master of the Aachen Altar), biographies, related people or corporate bodies, and other information about artists, architects, firms, studios, museums, special collections, patrons, donors, sitters, creating cultures (e.g., unknown Etruscan), and other people and groups involved in the creation, distribution, collection, maintenance, and study of art and architecture. Records in ULAN include either individuals (persons) or groups of individuals working together (corporate bodies). "Artists" in the ULAN generally represent creators involved in the conception or production of visual arts and architecture; performance artists are included, but typically ULAN does not focus on actors, dancers, or other performing artists.

Work on the ULAN began in 1984, when the Getty – informed by the terminology discussions of CIHA (Comité international d’histoire de l’art) – decided to merge and coordinate controlled vocabulary resources for use by the J. Paul Getty Trust's many automated documentation projects. In time, large numbers of contributions were submitted by other institutions involved in cataloging art and scholarship about art. In 1987 the Getty created a department dedicated to compiling and distributing terminology. The ULAN grows and changes via contributions from the user community and editorial work of the Getty Vocabulary Program.

The focus of each ULAN record (called a subject) is a person or group (artist, corporate body, patron, etc). Linked to each artist record are names, sources for the data, and notes. The temporal coverage of the ULAN ranges from Antiquity to the present and the scope is global.

Names in ULAN may include given names, pseudonyms, variant spellings, names in multiple languages, and names that have changed over time (e.g., married names). Among these names, one is flagged as the preferred name.

Even though the structure is relatively flat, the ULAN is constructed as a hierarchical thesaurus; compliant with ISO and NISO standards for thesaurus construction. It currently has five published facets (see ULAN Hierarchy and Classes): Persons, Artists; Corporate Bodies; Non-Artists; Unidentified Named People; and Unknown People by Culture. Entities in facets other than Corporate Bodies typically (but not always) have no children. Entities in the Corporate Bodies facet may branch into trees of Partitive relations. There may be multiple parents, making the ULAN structure polyhierarchical.

See more information about the history, purpose and scope of the ULAN; and see ULAN Specifics for the semantic representation of ULAN data.
1.2 Revisions, Review, Feedback

1.2.1 Revisions

We anticipate that in the future, the GVP LOD data will be refreshed every two weeks, on the same schedule as the online public data and the data files available to licensees through Web services. This document and the underlying ontology and mapping is also updated frequently.

- A record of all significant changes is in the following subsections
- Appropriate metadata (dct:modified, dct:issued, owl:versionInfo) is available in the Descriptive Information

1.2.1.1 Version 1.0

19 Feb 2014: Initial Version

1.2.1.2 Version 1.1

28 Feb 2014: Draft that was posted for about 2 weeks. Do not use.

1.2.1.3 Version 1.2

14 Mar 2014

- Mapping change: removed intermediate bibo:DocumentPart node in Language Tags and Sources
- Mapping change: removed Top Concepts indication since it is not considered useful
- Mapping change: touched Relationship Representation a bit to fit better the used ontology documentation tool (Parrot).
- Added comprehensive Descriptive Information (VOID etc)
- Added documentation: Language Tag Case, ontology documentation at end of GVP Ontology, clarification at end of Hierarchy Structure, clarification at end of Language Dual URLs.
- Sample Queries: added Language Queries (8), added Explore the Ontology (3), updated the queries in the SPARQL endpoint, provided link to this documentation (exclamation point icon)
- Site change: removed parasitic word "/resource" from GVP URLs

1.2.1.4 Version 1.3

15 Apr 2014

- Mapping change: omitted Sort Order of Facets and Hierarchies since it's not considered useful
- Omitted blank nodes from Per-Entity Exports and Total Exports (see query All Data For Subject)
- Added another illustration of the VOID domain model
- Added URLs for VOID Linksets, e.g. http://vocab.getty.edu/dataset/aat/alignment/lcsh
- Added the full VOID info to the repository (in addition to a designated file), see VOID Deployment. Changed the queries for Dynamic Descriptive Properties to INSERT instead of CONSTRUCT. Added all Prefixes to prefixes.ttl
- Added 2 descriptive info queries at the end of Counting and Descriptive Info
- Produced this document in HTML, in addition to PDF

1.2.1.5 Version 2.0

12 Aug 2014 (published 19 Aug 2014)

- Added the second vocabulary: TGN (see TGN Specifics and TGN-Specific Queries), official launch is expected 21 Aug.
- Mapping change: Made Term Characteristics independent of AAT (moved GVP URLs one level up), in preparation for adding TGN.
- Introduced "Extended" versions of GVP Hierarchical Relations, Infer Standard Hierarchical Relations (e.g. iso:broaderGeneric) from these Extended versions (e.g. gvp:broaderGeneric), see BTG, BTP, BTI Inference, ISO Insert Queries, ISO Rules.
- Added diagram Hierarchical Relations Inference that should clarify the different relations and which contributes to which others.
- Mapping change: replaced gvp:broadTransitive with gvp:broadExtended. For BTG & BTP (the only kinds of broader appearing in AAT and TGN), skos:broadTransitive is a restriction of gvp:broadExtended over skos:Concepts. But if/when BTI statements come into the picture, skos:broadTransitive makes some inappropriate inferences, so you should use gvp:broadExtended for appropriate query expansion.
- Mapping change: removed some superfluous relations, see struck-out properties in Standard Hierarchical Relations, GVP Hierarchical Relations and Reduced SKOS Inference. This is in order to reduce the number of triples (from 440M, representing an inference expansion ratio of 4x, to 162M and 1.6x). If you want them back, see Total Exports.
- Mapping change: omitted Undetermined language in GVP Language Tags.
- Mapping change: use schema:startDate,endDate for Historic Information and Obsolete Subject, instead of custom properties gvp:startDate,endDate.
- Mapping change: map Scope Note to custom class gvp:ScopeNote with rdf:value, instead of xl:Label with xl:literalForm.
- Declared Associative Relationships as sub-properties of skos:related (see Relationship Representation).
- Added more dct:formats to Descriptive Properties (both NTriples and ZIP for total exports, "meta/void" for descriptor, "meta/rdf-schema" for ontology).
- Added Named Graphs (used for a very limited purpose).
- Moved this document to a more logical location (see first page).
- Made the HTML version primary, and made properly named anchors for each section.

1.2.1.6 Version 2.1
12 Nov 2014
- Added VOID Descriptive Information for the GVP dataset, comprising AAT and TGN (previously we had AAT only).
- Corrected the Counting and Descriptive Info queries to take into account the different datasets.
- The GVP ontology is registered at Linked Open Vocabularies: http://lov.okfn.org/dataset/lov/details/vocabulary_gvp.html.
- Handle more Turtle MIME types in Semantic Resolution (text/n3, application/x-turtle).
- Provide the preferred Place Type in Full Text Search Query (e.g. "nation" or "settlement"), in addition to the concept type (e.g. AdminPlaceConcept).
- Query to get the Custom Language Tags adopted by GVP.

1.2.1.7 Version 3.0
30 Apr 2015
Mapping change:
- Explain the uncertain nature of GVP dates in Estimated Dates.
- Use gvp:estStart, estEnd instead of schema:startDate, endDate for Historic Information, Obsolete Subject and ULAN Events and Biographies.

ULAN:
- Added ULAN semantic representation: ULAN Specifics, various other places in the document and ULAN-Specific Queries.
- Updated http://vocab.getty.edu/ontology to include ULAN-specific classes and properties.
- Added LCNAF Alignment for ULAN.
Sample queries:
- Moved sample queries to a separate document http://vocab.getty.edu/doc/queries.
• Completely revamped the sample query UI in the endpoint, using the separate document as index

Other:
• Added Per-Resource Descriptive Info
• Fixed several illustrations to use properties going upwards (skos:broader, iso:superOrdinate) instead of downwards (skos:narrower, iso:subordinateArray), and to strike-out skos:isTopConceptOf
• Added hyperlink reference in W3C ADMS
• Added External Prefix "sesame:", which see

1.2.1.8 Version 3.1
5 June 2015.
No changes to the semantic mapping. Updated functionality:
• Added Twitter and Google forum updates to GVP LOD Home Page
• Made short URLs for online Subject & Hierarchy displays (see Common GVP URLs and Semantic Resolution), which was a long-standing request
• Added CSV and TSV Semantic Formats. These are very useful for some end-user tools like Excel and OpenRefine
• Fixed a bug where the SPARQL endpoint always assumed "Expand Equivalent URLs" (owl:sameAs) for output formats JSON or XML

Documentation changes:
• Changed identity of LCSH source (now there are two) used for LCSH Alignment
• Added clarifications to Resource View
• Added section Resource Titles and corresponding query Smart Resource Title (this was implemented in Sep 2014, now is extended and documented)

1.2.1.9 Version 3.2
15 Dec 2015.
No mapping changes. Functional improvements:
• Added JSONLD output format to Semantic Formats, including Per-Entity Exports and Query Results

VOID file improvements:
• Added all supported formats (a total of 9) as void:feature in Descriptive Properties
• Added the supported semantic formats (a total of 9) as void:uriRegexPattern for the different VOID Subsets
• Added void:uriLookupEndpoint to Descriptive Properties
• Added Wikidata identities for AAT, TGN, ULAN in void:theme in Descriptive Properties
• Described LCNAF Alignment in addition to LCSH Alignment in VOID Linksets
• Bumped version number to 3.2 in all datasets and the ontology (no changes since 3.0)
• Updated download sizes in Explicit Exports, Total Exports and the VOID file
• Switched to MIME type URL http://www.iana.org/assignments/media-types/application/zip since both http://purl.org/NET/mediatypes/application/zip and http://provenanceweb.org/format/mime/application/zip do not resolve anymore

Documentation improvements:
• Put figure ADMS on a line of its own.
• Updated paper reference in GVP Hierarchical Relations and BTG, BTP, BTI Inference to official published version.
• Added link to query ULAN Events by Type in ULAN Life Events
• Split out section Semantic Formats
• Fixed some broken links
1.2.1.10 **Version 3.3**
20 May 2016

Ontology addition:
- Added new value for gvp:termKind <term/kind/Code> to the ontology
- Added rdfs:isDefinedBy to **Relationship Representation**

Documentation improvements:
- Fixed Vapour link in **Semantic Resolution**
- Provided link to The rationale of PROV in PROV

1.2.1.11 **Version 3.4**

- Added freq: (DC Collection Description Frequency) and removed voag: from **Descriptive Prefixes**: VOAG (and other ontologies describing frequency of update) have become unavailable
- Switched property dct:accrualPeriodicity from value voag:BiWeekly to freq:biweekly in **Descriptive Properties**. Removed property voag:frequencyOfChange.

Documentation improvements:
- Fixed 20 links (out of over 1200)
- Added link to Google Sheet in IANA Language Tags
- Clarified status of RKBExplorer VOID in **VOID Deployment**
- Updated download file sizes in Explicit Exports and Total Exports

1.2.1.12 **Future Versions**

Your feedback (positive/negative and relative importance) is appreciated on the following amendments under consideration. See the corresponding sections for details.

Semantic Additions:
- Map TGN to more **Geographic Ontologies** (which ones?)
- Map ULAN to CIDOC CRM
- Map **TGN Place Types** to gvp:broaderInstantial and/or make a hierarchical version (gvp:placeTypeExtended or gvp:broaderInstantialExtended)
- Omit gvp:historicFlag=<historic/current> from **Historic Information** and adopt that as default
- Introduce special lang tags @x-iso6392, @x-iso6393 (and @x-iana) to mark language codes with, instead of English
- Publish images of AAT concepts or images related to ULAN artists as foaf:depiction
- Consider a sub-property hierarchy in the **Associative Relationships**
- Add back gvp:narrower and skos:narrower that were removed in version 2.0 (but not their transitive variants). Let us know if you need them.

Descriptive Information:
- Provide **SPARQL 1.1 Service Description** (SD) of the endpoint, tie up the VOID with SD.
- Provide an additional **VOID Deployment** method through the endpoint URL, as per **SPARQL 1.1 Service Description section 2**
- Provide an **Open Search 1.1** description document in void:openSearchDescription, to allow applications to discover the Full Text Search automatically
- Provide Vocabulary of a Friend (VOAF) metadata and submit the GVP ontology to the **Linked Open Vocabularies** (LOV) site
- Provide **VOID-ext** metadata, in particular vext:languagePartition and vext:languages
- Consider VAEM, VDPP and VOAG metadata (but we are not convinced they have significant penetration )

Functional and Endpoint Additions:
- Provide Advanced Search (e.g. concepts/places by parent, places by type) in addition to the **Full Text Search**
• Provide Map views of TGN places and search results

1.2.2 External Review Process

Numerous individuals are serving as External Advisors on this project. Most have been recommended by colleagues in our community (e.g., International Terminology Working Group members who are currently translating the AAT into various languages). We ended up inviting a fairly large group because we wanted to make sure that we had expertise in many areas. It has been very important to us that these trusted colleagues had a chance to comment on our ontology choices prior to the release of the datasets.

1.2.3 Providing Feedback

We welcome comments if you find something in this document or our dataset that needs clarification or improvement. We ask specifically for your opinion in several places in this document, as listed in Future Versions. We have established a public discussion forum (Google Group) which we hope the community will use to ask questions, discuss issues, and find solutions related to the technical aspects of this publication.

• Explore: Getty Vocabularies as Linked Open Data
• Join: Send email to gettylvocablod+subscribe@googlegroups.com or by visiting https://www.googlegroups.com, searching for ‘Getty Vocabularies’, and using the ‘Join Group’ option.

Previously we provided support at http://answers.semanticweb.com/tags/getty. Unfortunately this site is gone, but is last archived in Oct 2015 (17 answers).

Questions and comments about editorial content or general information regarding the Getty Vocabularies (not LOD) should be directed to vocab@getty.edu.

1.2.4 Disclaimer

The vocabulary datasets are provided "as is". The Getty disclaims all other warranties, either express or implied, including, but not limited to, implied warranties of merchantability and fitness for a particular purpose, with respect to the database. The Getty Vocabularies are compiled by the Getty Vocabulary Program from contributions from various contributors, including museums, libraries, archives, bibliographic indexing projects, international translation projects, and others. Not all contributor data complies precisely with the GVP Editorial Guidelines; therefore, absolute consistency in the dataset is not possible. The data is subject to frequent (biweekly) updates and corrections. Please keep in mind that the ULAN portion of the data is considered by the Getty to be a BETA release. It is being offered early to give the external community a chance to comment prior to the official launch date in late April 2015.

• LOD data is provided as Export Files
• See more information about various ways in which the Getty vocabularies may be obtained.
### 1.3 Abbreviations

The following abbreviations are used in the document. In addition, [External Prefixes](#) and [Descriptive Prefixes](#) includes abbreviations of the ontologies used (e.g. RDF is Resource Description Framework, VoID is Vocabulary of Interlinked Datasets, etc)

<table>
<thead>
<tr>
<th>Abbrev</th>
<th>Term</th>
</tr>
</thead>
<tbody>
<tr>
<td>AACR2</td>
<td>Anglo-American Cataloging Rules 2</td>
</tr>
<tr>
<td>AAT</td>
<td>Art and Architecture Thesaurus</td>
</tr>
<tr>
<td>AATNed</td>
<td>AAT Netherlands: the project to make the Dutch translation of the AAT</td>
</tr>
<tr>
<td>BTG</td>
<td>Broader Than Generic (Genus/Species) relation: property broaderGeneric</td>
</tr>
<tr>
<td>BTP</td>
<td>Broader Than Partitive (Part/Whole) relation: property broaderPartitive</td>
</tr>
<tr>
<td>BTI</td>
<td>Broader Than Instantial (Kind/Instance) relation: property broaderInstantial</td>
</tr>
<tr>
<td>CCO</td>
<td>Cataloging Cultural Objects</td>
</tr>
<tr>
<td>CDWA</td>
<td>Categories for the Description of Works of Art</td>
</tr>
<tr>
<td>CSV</td>
<td>Comma-Separated Values</td>
</tr>
<tr>
<td>Forest</td>
<td>Ontotext semantic UI framework</td>
</tr>
<tr>
<td>FTS</td>
<td>Full Text Search</td>
</tr>
<tr>
<td>GCI</td>
<td>Getty Conservation Institute</td>
</tr>
<tr>
<td>GRI</td>
<td>Getty Research Institute</td>
</tr>
<tr>
<td>GVP</td>
<td>Getty Vocabulary Program</td>
</tr>
<tr>
<td>IANA</td>
<td>Internet Assigned Numbers Authority</td>
</tr>
<tr>
<td>ISO</td>
<td>International Standardization Organization</td>
</tr>
<tr>
<td>LCSH</td>
<td>Library of Congress Subject Headings</td>
</tr>
<tr>
<td>LCNAF</td>
<td>Library of Congress Name Authority File</td>
</tr>
<tr>
<td>LoC</td>
<td>Library of Congress</td>
</tr>
<tr>
<td>LOD</td>
<td>Linked Open Data</td>
</tr>
<tr>
<td>Lucene</td>
<td>Lucene FTS engine</td>
</tr>
<tr>
<td>OWLIM</td>
<td>Ontotext semantic (RDF) repository, former name of GraphDB</td>
</tr>
<tr>
<td>RDFa</td>
<td>RDF in Attributes: allows the embedding of RDF data in HTML</td>
</tr>
<tr>
<td>SPARQL</td>
<td>SPARQL Protocol and RDF Query Language</td>
</tr>
<tr>
<td>TGN</td>
<td>Thesaurus of Geographic Names</td>
</tr>
<tr>
<td>TSV</td>
<td>Tab-Separated Values</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>ULAN</td>
<td>Union List of Artist Names</td>
</tr>
<tr>
<td>URI</td>
<td>Unified Resource Identifier. Following LOD principles, all our URIs are resolvable, i.e. URLs</td>
</tr>
<tr>
<td>URL</td>
<td>Unified Resource Locator</td>
</tr>
<tr>
<td>W3C</td>
<td>World Wide Web Consortium</td>
</tr>
</tbody>
</table>
1.4 RDF Turtle

This document uses examples written in Turtle, which is much more readable than other RDF representations (see Semantic Formats). We use URLs derived from AAT database IDs, so our prefixed names start with a digit.

Examples:
- aat:300198841: a Subject
- aat_term:1000198841-el-Latn: a Term
- aat_source:2000051089-term-1000198841: a Source (namely the "local source") pertaining to this term

The Turtle 1.1 Candidate Recommendation of 19 February 2013 allows the local part of a prefixed name to start with a digit.

We provide Export Files in Turtle, RDF XML, NTriples, JSON, JSONLD. To load the Turtle files you need recent parsing tools that support this Turtle 1.1 feature. For example:
- Sesame RIO 2.7.9 (2013-12-18)
- Jena ARQ 2.8.8 (2011-04-21)

Turtle 1.1 names may also include colon ":" (CURIE) and may include other special characters through escaping (e.g. at-sign "@" and slash "/"). However we limit the characters used in local names to letters, digits and dash "-" (e.g. the Jena ARQ version cited above does not handle ":" and escaped chars). Because Turtle 1.1 capable tools are not very widely deployed as of the end of 2013, we provide the Total Export files in NTriples format.

1.5 Prefixes

The prefixes that we use (both internal and external) are defined in the following sections.

The easiest way to find prefixes is the prefix.cc service.
- For example, if you wonder what the rr: prefix means, go to http://prefix.cc/rr. You will quickly find the URL, and whether any other alternatives have been registered.
- For the GVP Prefixes, we have registered gvp: aat: tgn: and ulan: in this service (unfortunately it does not support prefixes including "_")
- If you need the prefix to put in a Turtle file, go directly to e.g. http://prefix.cc/rr.ttl. This page allows you to copy it to the clipboard, so you can paste it into a file
- You can request several prefixes at once, e.g.: http://prefix.cc/gvp,aat,tgn,ulan.ttl

For your convenience, all prefixes that we use are provided at http://vocab.getty.edu/doc/prefixes.ttl (formerly http://www.getty.edu/research/tools/vocabularies/lod/prefixes.ttl)
1.5.1 External Prefixes

We use the following prefixes for External Ontologies.

- **rr**: and **rrx**: (R2RML) are used only during the conversion process.
- **luc**: is used for Full Text Search Query.

<table>
<thead>
<tr>
<th>Prefix</th>
<th>URL</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>bibo:</td>
<td><a href="http://purl.org/ontology/bibo/">http://purl.org/ontology/bibo/</a></td>
<td>Bibliography Ontology</td>
</tr>
<tr>
<td>bio:</td>
<td><a href="http://purl.org/vocab/bio/0.1/">http://purl.org/vocab/bio/0.1/</a></td>
<td>Biography Ontology</td>
</tr>
<tr>
<td>dc:</td>
<td><a href="http://purl.org/dc/elements/1.1/">http://purl.org/dc/elements/1.1/</a></td>
<td>Dublin Core Elements</td>
</tr>
<tr>
<td>dct:</td>
<td><a href="http://purl.org/dc/terms/">http://purl.org/dc/terms/</a></td>
<td>Dublin Core Terms</td>
</tr>
<tr>
<td>foaf:</td>
<td><a href="http://xmlns.com/foaf/0.1/">http://xmlns.com/foaf/0.1/</a></td>
<td>Friend of a Friend ontology</td>
</tr>
<tr>
<td>iso:</td>
<td><a href="http://purl.org/iso25964/skos-thes#">http://purl.org/iso25964/skos-thes#</a></td>
<td>ISO 25946 ontology (latest ISO standard on thesauri)</td>
</tr>
<tr>
<td>luc:</td>
<td><a href="http://www.ontotext.com/owlim/lucene#">http://www.ontotext.com/owlim/lucene#</a></td>
<td>Ontotext GraphDB's built-in Lucene Full Text Search</td>
</tr>
<tr>
<td>ontogeo:</td>
<td><a href="http://www.ontotext.com/owlim/geo#">http://www.ontotext.com/owlim/geo#</a></td>
<td>Ontotext GraphDB geo-spatial extensions, e.g. Places Within Bounding Box, Places Nearby Each Other</td>
</tr>
<tr>
<td>owl:</td>
<td><a href="http://www.w3.org/2002/07/owl#">http://www.w3.org/2002/07/owl#</a></td>
<td>Web Ontology Language</td>
</tr>
<tr>
<td>prov:</td>
<td><a href="http://www.w3.org/ns/prov#">http://www.w3.org/ns/prov#</a></td>
<td>Provenance Ontology</td>
</tr>
<tr>
<td>ptop:</td>
<td><a href="http://www.ontotext.com/proton/protontop#">http://www.ontotext.com/proton/protontop#</a></td>
<td>PROTON ontology, used in Extended Property Constructs</td>
</tr>
<tr>
<td>rdf:</td>
<td><a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#">http://www.w3.org/1999/02/22-rdf-syntax-ns#</a></td>
<td>Resource Description Framework</td>
</tr>
<tr>
<td>rdfs:</td>
<td><a href="http://www.w3.org/2000/01/rdf-schema#">http://www.w3.org/2000/01/rdf-schema#</a></td>
<td>RDF Schema</td>
</tr>
<tr>
<td>rr:</td>
<td><a href="http://www.w3.org/ns/r2rml#">http://www.w3.org/ns/r2rml#</a></td>
<td>Relational to RDF Mapping Language, used for conversion only</td>
</tr>
<tr>
<td>rrx:</td>
<td><a href="http://purl.org/r2rml-ext/">http://purl.org/r2rml-ext/</a></td>
<td>R2RML extension (rrx:languageColumn)</td>
</tr>
<tr>
<td>schema:</td>
<td><a href="http://schema.org/">http://schema.org/</a></td>
<td>Schema.org common properties</td>
</tr>
<tr>
<td>sesame:</td>
<td><a href="http://www.openrdf.org/schema/sesame#">http://www.openrdf.org/schema/sesame#</a></td>
<td>Special predicate directSubPropertyOf (see query Associative Relations of Agent)</td>
</tr>
<tr>
<td>skos:</td>
<td><a href="http://www.w3.org/2004/02/skos/core#">http://www.w3.org/2004/02/skos/core#</a></td>
<td>Simple Knowledge Organization System</td>
</tr>
<tr>
<td>skosxl:</td>
<td><a href="http://www.w3.org/2008/05/skos-xl#">http://www.w3.org/2008/05/skos-xl#</a></td>
<td>SKOS Extension for Labels</td>
</tr>
<tr>
<td>wgs:</td>
<td><a href="http://www.w3.org/2003/01/geo/wgs84_pos#">http://www.w3.org/2003/01/geo/wgs84_pos#</a></td>
<td>W3C Geo ontology (WGS stands for the World Geodetic Survey 1984 datum)</td>
</tr>
<tr>
<td>xsd:</td>
<td><a href="http://www.w3.org/2001/XMLSchema#">http://www.w3.org/2001/XMLSchema#</a></td>
<td>XML Schema Datatypes</td>
</tr>
</tbody>
</table>
1.5.2 Descriptive Prefixes

We use some of the following prefixes for **Descriptive Information**:

<table>
<thead>
<tr>
<th>Prefix</th>
<th>URL</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>adms:</td>
<td><a href="http://www.w3.org/ns/adms#">http://www.w3.org/ns/adms#</a></td>
<td>Asset Description Metadata Scheme</td>
</tr>
<tr>
<td>cc:</td>
<td><a href="http://creativecommons.org/ns#">http://creativecommons.org/ns#</a></td>
<td>Creative Commons Rights Expression Language</td>
</tr>
<tr>
<td>dc:</td>
<td><a href="http://purl.org/dc/elements/1.1/">http://purl.org/dc/elements/1.1/</a></td>
<td>Dublin Core Metadata Element Set</td>
</tr>
<tr>
<td>dcat:</td>
<td><a href="http://www.w3.org/ns/dcat#">http://www.w3.org/ns/dcat#</a></td>
<td>Data Catalog Vocabulary</td>
</tr>
<tr>
<td>dct:</td>
<td><a href="http://purl.org/dc/terms/">http://purl.org/dc/terms/</a></td>
<td>DCMI Metadata Terms</td>
</tr>
<tr>
<td>dctype:</td>
<td><a href="http://purl.org/dc/dcmitype/">http://purl.org/dc/dcmitype/</a></td>
<td>DCMI Type Vocabulary</td>
</tr>
<tr>
<td>fmt:</td>
<td><a href="http://www.w3.org/ns/formats/">http://www.w3.org/ns/formats/</a></td>
<td>RDF formats used in datasets</td>
</tr>
<tr>
<td>freq:</td>
<td><a href="http://purl.org/cld/freq/">http://purl.org/cld/freq/</a></td>
<td>DC Collection Description terms: frequency</td>
</tr>
<tr>
<td>sd:</td>
<td><a href="http://www.w3.org/ns/sparql-service-description#">http://www.w3.org/ns/sparql-service-description#</a></td>
<td>SPARQL Service Description</td>
</tr>
<tr>
<td>vaem:</td>
<td><a href="http://www.linkedmodel.org/schema/vaem#">http://www.linkedmodel.org/schema/vaem#</a></td>
<td>Vocabulary for Attaching Essential Metadata</td>
</tr>
<tr>
<td>vann:</td>
<td><a href="http://purl.org/vocab/vann/">http://purl.org/vocab/vann/</a></td>
<td>Vocabulary for annotating vocabulary descriptions</td>
</tr>
<tr>
<td>vcard:</td>
<td><a href="http://www.w3.org/2006/vcard/ns#">http://www.w3.org/2006/vcard/ns#</a></td>
<td>vCard (contact info)</td>
</tr>
<tr>
<td>vdpp:</td>
<td><a href="http://data.lirmm.fr/ontologies/vdpp#">http://data.lirmm.fr/ontologies/vdpp#</a></td>
<td>Vocabulary for Dataset Publication Projects</td>
</tr>
<tr>
<td>voaf:</td>
<td><a href="http://purl.org/voccommons/voaf#">http://purl.org/voccommons/voaf#</a></td>
<td>Vocabulary of a Friend</td>
</tr>
<tr>
<td>void:</td>
<td><a href="http://rdfs.org/ns/void#">http://rdfs.org/ns/void#</a></td>
<td>Vocabulary of Interlinked Datasets</td>
</tr>
<tr>
<td>wdrs:</td>
<td><a href="http://www.w3.org/2007/05/powder-s#">http://www.w3.org/2007/05/powder-s#</a></td>
<td>Protocol for Web Description Resources</td>
</tr>
<tr>
<td>wv:</td>
<td><a href="http://vocab.org/waiver/terms/">http://vocab.org/waiver/terms/</a></td>
<td>A vocabulary for waivers of rights</td>
</tr>
</tbody>
</table>

1.6 GVP URLs and Prefixes

The layout of GVP URLs (both ontology and vocabulary entities) is shown below.

- We use a template notation in the URLs: `{voc}` indicates a vocabulary, `{m}` indicates a numeric identifier, `{v}` some value, `{lang}` a language tag.
- We have defined prefixes for the most important URLs.
- The AAT subject URIs `aat:{m}` are the only ones that will be used in external datasets, so they are more important than all the other URIs. We keep them as short as possible, allowing shortest Turtle and SPARQL.
- We have adopted a "dual approach" for the TGN & ULAN subject URIs. The concept URI is the same form as AAT: `tgn:{m}` and `ulan:{m}`. Concepts are the "business objects" (records) of thesaurus management systems and the daily business of editorial teams such as the GVP. Things, on the other hand, exist (or have existed) independently in the real world. For these we use `tgn:{m}-place` and `ulan:{m}-agent`. In your data you will most often want to use the thing URIs, see [Concept vs Thing Duality](#).
- URLs shown in **roman font** represent independent entities. The data of URLs shown in *italic font* is returned together with the owning Subject so that you won't have to chase different URLs to gather the information (these URLs can also be resolved on their own).
### 1.6.1 Common GVP URLs

<table>
<thead>
<tr>
<th>Prefix</th>
<th>URL</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>base</td>
<td><a href="http://vocab.getty.edu/">http://vocab.getty.edu/</a></td>
<td>Prepend to all URIs. Returns a home page describing the LOD vocabularies and giving links to the ontology, vocabularies, sample resources, etc</td>
</tr>
<tr>
<td>page/{voc}/{m}</td>
<td></td>
<td>Human-readable page about subject {m}</td>
</tr>
<tr>
<td>hier/{voc}/{m}</td>
<td></td>
<td>Human-readable page showing the hierarchical position of subject {m}</td>
</tr>
<tr>
<td>dataset/</td>
<td></td>
<td>Directories with data dumps: (aat</td>
</tr>
<tr>
<td>gvp:</td>
<td>ontology#</td>
<td>GVP Ontology. Uses SKOS, SKOS-XL, ISO 25964, DC, DCT, BIBO, FOAF and PROV. Also holds Associative Relations</td>
</tr>
<tr>
<td>gvp_lang:</td>
<td>language/{lang}</td>
<td>Languages used by GVP. URLs use IANA language tags. Have owl:sameAs links to corresponding concepts in the AAT Languages hierarchy</td>
</tr>
<tr>
<td>historic/{v}</td>
<td></td>
<td>Historic Flag: Current, Historic or Both</td>
</tr>
<tr>
<td>term/display/{v}</td>
<td></td>
<td>Term use: for Display or Indexing?</td>
</tr>
<tr>
<td>term/flag/{v}</td>
<td></td>
<td>Term Flag: Vernacular, Loan Term</td>
</tr>
<tr>
<td>term(kind/{v}</td>
<td></td>
<td>Term Kind: Neologism, Scientific Name, etc</td>
</tr>
<tr>
<td>term/POS/{v}</td>
<td></td>
<td>Term Part of Speech: Noun Plural, Noun Singular, etc</td>
</tr>
<tr>
<td>term/type/{v}</td>
<td></td>
<td>Term Type: Descriptor, Alternate Descriptor, Use For term</td>
</tr>
<tr>
<td>.well-known/void</td>
<td></td>
<td>VOID Descriptive Information (see VOID Deployment)</td>
</tr>
<tr>
<td>void.ttl</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 1.6.2 AAT URLs

<table>
<thead>
<tr>
<th>Prefix</th>
<th>URL</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>aat:</td>
<td>aat/</td>
<td>AAT vocabulary (skos:ConceptScheme)</td>
</tr>
<tr>
<td>aat/{m}</td>
<td></td>
<td>AAT subject {m}: Facet, Hierarchy, Guide Term, or Concept</td>
</tr>
<tr>
<td>aat/{m}-array</td>
<td></td>
<td>Anonymous iso:ThesaurusArray used to represent the ordered concept {m}</td>
</tr>
<tr>
<td>aat/{m}-list-{n}</td>
<td></td>
<td>rdf:List element for child subject {n} of the skos:OrderedCollection used to represent the ordered subject {m}</td>
</tr>
<tr>
<td>aat_contrib:</td>
<td>aat/contri/{m}</td>
<td>AAT Contributor {m}</td>
</tr>
<tr>
<td>aat_rel:</td>
<td>aat/rel/{m}-type-{n}</td>
<td>AAT Relation: from subject {m} to subject {n}, having {type} (&quot;broader&quot; or an associative relation)</td>
</tr>
<tr>
<td>aat_rev:</td>
<td>aat/rev/{m}</td>
<td>AAT Revision of a subject</td>
</tr>
<tr>
<td>aat_scopeNote:</td>
<td>aat/scopeNote/{m}</td>
<td>AAT Scope Note (definition) {m}</td>
</tr>
<tr>
<td>aat_source:</td>
<td>aat/source/{m}</td>
<td>AAT Source {m}</td>
</tr>
<tr>
<td>aat/source/{m}-scopeNote-{n}</td>
<td></td>
<td>AAT Source {m} applied to scope note {n}</td>
</tr>
<tr>
<td>aat/source/{m}-subject-{n}</td>
<td></td>
<td>AAT Source {m} applied to subject {n}</td>
</tr>
<tr>
<td>aat/source/{m}-term-{n}</td>
<td></td>
<td>AAT Source {m} applied to term {n}</td>
</tr>
<tr>
<td>aat_term:</td>
<td>aat/term/{m}</td>
<td>AAT Term {m}</td>
</tr>
</tbody>
</table>

Sources and Contributors are thersaurus-dependent, e.g. aat_contrib:123 and tgn_contrib:123 are different records. Even if the same organization contributes to both AAT and TGN, it will get two different URLs. Contributors are also not correlated to ULAN.
# Getty Vocabularies: LOD

## Semantic Representation

### 1.6.3 TGN URLs

<table>
<thead>
<tr>
<th>Prefix</th>
<th>URL</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>tgn:</td>
<td>tgn/</td>
<td>TGN vocabulary (skos:ConceptScheme)</td>
</tr>
<tr>
<td>tgn/{m}</td>
<td></td>
<td>TGN subject {m}: Facet, gvp:PhysPlaceConcept, gvp:AdminPlaceConcept, or gvp:PhysAdminPlaceConcept</td>
</tr>
<tr>
<td>tgn/{m}-place</td>
<td></td>
<td>Place (wgs: SpatialThing and schema: Place) corresponding to TGN concept /tgn/{m} (see Concept vs Thing Duality)</td>
</tr>
<tr>
<td>tgn/{m}-geometry</td>
<td></td>
<td>Geometry (schema:GeoCoordinates, schema:GeoShape) of /tgn/{m}-place. Schema coordinates are in this node, but WGS coordinates are in /tgn/{m}-place.</td>
</tr>
<tr>
<td>tgn/{m}-array</td>
<td></td>
<td>Anonymous iso: ThesaurusArray used to represent the ordered concept {m}</td>
</tr>
<tr>
<td>tgn/{m}-list-{n}</td>
<td></td>
<td>rdf: List element for child subject {n} of the skos: OrderedCollection used to represent the ordered concept {m}</td>
</tr>
<tr>
<td>tgn_contrib:</td>
<td>tgn/contrib/{m}</td>
<td>TGN Contributor {m}</td>
</tr>
<tr>
<td>tgn_rel:</td>
<td>tgn/rel/{m}-{type}-{n}</td>
<td>TGN Relation {m}-{type}-{n}: from subject {m} to subject {n}, having {type} (&quot;broader&quot;, &quot;placeType&quot;, or an associative relation)</td>
</tr>
<tr>
<td>tgn_rev:</td>
<td>tgn/rev/{m}</td>
<td>TGN Revision of a subject</td>
</tr>
<tr>
<td>tgn_scopeNote:</td>
<td>tgn/scopeNote/{m}</td>
<td>TGN Scope Note (definition) {m}</td>
</tr>
<tr>
<td>tgn_source:</td>
<td>tgn/source/{m}</td>
<td>TGN Source {m}</td>
</tr>
<tr>
<td>tgn/source/{m}-scopeNote-{n}</td>
<td></td>
<td>TGN Source {m} applied to scope note {n}</td>
</tr>
<tr>
<td>tgn/source/{m}-subject-{n}</td>
<td></td>
<td>TGN Source {m} applied to subject {n}</td>
</tr>
<tr>
<td>tgn/source/{m}-term-{n}</td>
<td></td>
<td>TGN Source {m} applied to term {n}</td>
</tr>
<tr>
<td>tgn_source_rev:</td>
<td>tgn/source/rev/{m}</td>
<td>TGN Revision of a Source</td>
</tr>
<tr>
<td>tgn_term:</td>
<td>tgn/term/{m}</td>
<td>TGN Term {m}</td>
</tr>
</tbody>
</table>

### 1.6.4 ULAN URLs

<table>
<thead>
<tr>
<th>Prefix</th>
<th>URL</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ulan:</td>
<td>ulan/</td>
<td>ULAN vocabulary (skos:ConceptScheme)</td>
</tr>
<tr>
<td>ulan/{m}</td>
<td></td>
<td>ULAN subject {m}: Facet, GuideTerm, PersonConcept, GroupConcept, UnknownPersonConcept, ObsoleteSubject</td>
</tr>
<tr>
<td>ulan/{m}-agent</td>
<td></td>
<td>Agent (schema: Person or schema: Organization) corresponding to ULAN concept /ulan/{m} (see Concept vs Thing Duality)</td>
</tr>
<tr>
<td>ulan/{m}-array</td>
<td></td>
<td>Anonymous iso: ThesaurusArray used to represent the ordered GroupConcept {m} (if the children of an organization need to be ordered)</td>
</tr>
<tr>
<td>ulan/{m}-list-{n}</td>
<td></td>
<td>rdf: List element for child subject {n} of the skos: OrderedCollection used to represent the ordered GroupConcept {m}</td>
</tr>
<tr>
<td>ulan/{m}-nationality-{n}</td>
<td></td>
<td>rdf: Statement about nationality {n} of ULAN Subject {m} (includes only gvp: displayOrder)</td>
</tr>
<tr>
<td>ulan_bio:</td>
<td>ulan/bio/{m}</td>
<td>ULAN Biography (gvp: Biography)</td>
</tr>
<tr>
<td>ulan_contrib:</td>
<td>ulan/contrib/{m}</td>
<td>ULAN Contributor {m}</td>
</tr>
<tr>
<td>ulan_event:</td>
<td>ulan/event/{m}</td>
<td>ULAN Life Event (bio: Event, schema: Event)</td>
</tr>
<tr>
<td>Prefix</td>
<td>URL</td>
<td>Explanation</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ulan_rel</td>
<td>ulan/rel/{m}-[type]-{n}</td>
<td>ULAN Relation {m}-[type]-{n}: from subject {m} to subject {n}, having {type} (&quot;broader&quot;, &quot;placeType&quot;, or an associative relation)</td>
</tr>
<tr>
<td>ulan_rev</td>
<td>ulan/rev/{m}</td>
<td>ULAN Revision of a subject</td>
</tr>
<tr>
<td>ulan_scopeNote</td>
<td>ulan/scopeNote/{m}</td>
<td>ULAN Scope Note (definition) {m}</td>
</tr>
<tr>
<td>ulan_source</td>
<td>ulan/source/{m}</td>
<td>ULAN Source {m}</td>
</tr>
<tr>
<td>ulan_source/scopeNote-{n}</td>
<td>ulan/source/{m}-scopeNote-{n}</td>
<td>ULAN Source {m} applied to scope note {n}</td>
</tr>
<tr>
<td>ulan_source/subject-{n}</td>
<td>ulan/source/{m}-subject-{n}</td>
<td>ULAN Source {m} applied to subject {n}</td>
</tr>
<tr>
<td>ulan_source/term-{n}</td>
<td>ulan/source/{m}-term-{n}</td>
<td>ULAN Source {m} applied to term {n}</td>
</tr>
<tr>
<td>ulan_source_rev</td>
<td>ulan/source/rev/{m}</td>
<td>ULAN Revision of a Source</td>
</tr>
<tr>
<td>ulan_term</td>
<td>ulan/term/{m}</td>
<td>ULAN Term {m}</td>
</tr>
</tbody>
</table>

Please note that URLs including the parasitic word "/resource" (e.g. http://vocab.getty.edu/resource/aat/{m}) are NOT valid GVP URLs and you should not use them in your applications, nor make statements about them. This word was used in earlier versions of the http://vocab.getty.edu website due to the internal architecture of Forest UI. People were copying them from the browser address bar and sharing them in discussions, so we went to the trouble of reworking Forest to remove this word, in order to avoid confusion.

1.6.5 Using GVP URLs

GVP URLs are guaranteed to stay stable, as explained in Identifiers. If a subject is discontinued or merged to another, it'll be present in the GVP LOD as Obsolete Subject for at least 5 years.

- For AAT, you'll probably only want to use the Subject URLs aat:{m}
- For TGN, you should use the Place URL tgn:{m}-place in your Cultural Heritage data (e.g. as place of creation of a CH artifact or place of birth of a painter). An event cannot take place in a skos:Concept, so an appropriately typed node is needed for this purpose (see Concept vs Thing Duality).
- For ULAN, you should use the Agent URL ulan:{m}-agent in your Cultural Heritage data (e.g. as creator of an object). A concept cannot create an object, a person can.
- You may also want to use a specific label aat_term:{m} or tgn_term:{m}, if you are describing the use of a particular term/name in history.

1.6.6 Named Graphs

We split the GVP data in different named graphs:

- http://vocab.getty.edu/dataset:aat: AAT data (subjects, terms, revisions, relations, sources, contributors)
- http://vocab.getty.edu/dataset:tgn: TGN data (subjects, terms, revisions, relations, sources, contributors, places, geometries)
- http://vocab.getty.edu/dataset:ulan: ULAN data (subjects, terms, revisions, relations, sources, contributors, agents, nationalities, biographies, events)
- http://vocab.getty.edu/.well-known/void: Descriptive Information
- Default (empty) graph: a union of all the above, plus all inferred statements (see Inference)

We do this only so we can provide per-vocabulary triple counts in Dynamic Descriptive Properties. For most purposes, you should ignore the existence of these named graphs:

- You can get all data of a particular vocabulary using the separate Export Files.
- The named graphs are not available in the export files.
- You cannot get per-vocabulary data with filtering by named graph, since all inferred statements are in the empty graph (due to a limitation of Ontotext GraphDB).
- It would not be smart to deploy only TGN data, since TGN refers to AAT (TGN Place Types are AAT subjects)
You should follow these patterns:

- To get all subjects from a particular vocabulary, filter by skos:inScheme
- To get all data about a subject, use the Per-Entity Exports or the query All Data For Subject
- To get all sources or contributors for a particular vocabulary: Find Contributors by Vocabulary

### 1.7 Semantic Resolution

All GVP, AAT, TGN, and ULAN URLs resolve, returning human or machine readable content.

- We followed the recommendation Cool URLs for the Semantic Web
- We followed Best Practice Recipes for Publishing RDF Vocabularies
- We validated the resolution with Vapour (source location).

Different output formats can be obtained through:

- Content negotiation: use the Accept request header with the MIME type listed below
- Direct URL: use the URL with file extension, as shown in Example URLs.

#### 1.7.1 Semantic Formats

RDF is a graph data model that can be represented in a number of concrete data formats. Similarly, tabular SPARQL results can be transmitted in a number of formats. All these formats, and some other formats related to the semantic web, are described at [http://www.w3.org/ns/formats/](http://www.w3.org/ns/formats/) and linked pages (the usual prefix for this URL is fmt).

GVP LOD supports the most important semantic formats. We list them below, with link to specification, file extension, MIME type and brief (and subjective) notes.

**Human readable:**

- HTML (application/xhtml+xml): we provide 3 pages for the most important resources (see next section). In the future it would be nice to merge RDF and HTML using e.g. RDFa, let us know if this is something you need

**Semantic resources and semantic SPARQL formats (CONSTRUCT/ DESCRIBE queries):**

- RDF/XML (.rdf, application/rdf+xml): the oldest RDF format, it is mandated by several specifications but is also the hardest to read, and quite hard to process (because the same RDF can be expressed in many different RDF/XML forms)
- Turtle (.ttl, text/turtle): the most readable format.
- N-Triples (.nt, application/n-triples): a simple line-oriented format that's easy to process with Unix command-line tools.
- RDF/JSON (.json or .rj, application/rdf+json): an old JSON format that is not used much anymore.
- JSONLD (.jsonld, application/ld+json; also see home page): a more modern format that's easier to consume by web applications. It is especially important in the context of IIIF, which is an image interoperability and annotation framework that is popular in the Cultural Heritage domain. Unfortunately, the current GVP LOD implementation uses full URLs and not prefixes defined in a JSONLD Context. The reason are 6 pending Sesame subtasks of [SES-1094](https://github.com/metadigit/SES/pull/1094). If you need JSONLD Context immediately, please let us know.

**Tabular SPARQL formats (SELECT/ASK queries):**

- SPARQL XML (.xml or .srx, application/sparql-results+xml): supported by most SPARQL client frameworks
- SPARQL JSON (.json or .srj, application/sparql-results+json): supported by most SPARQL client frameworks, easier to parse by web applications
- SPARQL CSV (.csv, text/csv: comma separated values): useful for some end-user tools like Excel and OpenRefine.
- SPARQL TSV (.tsv, text/tab-separated-values): useful for some end-user tools like Excel and OpenRefine.

#### 1.7.2 Example URLs

For the direct URLs we use suffixes (file extensions) instead of prefixes (folders), since such URIs:

- Emphasize they all are about one semantic entity, instead of being put in "different folders", see Hierarchical URIs Pattern
- Are more hackable (easier to add suffix than spot the prefix), see Patterned URIs Pattern
- Provide correct file extensions when the file is downloaded
Example about the ontology:
- http://vocab.getty.edu/ontology: semantic URI, content-negotiated using 303 redirect
- http://vocab.getty.edu/ontology.rdf: RDF/XML
- http://vocab.getty.edu/ontology.ttl: Turtle

Example about an AAT subject: aventurine (quartz)
- http://vocab.getty.edu/page/aat/300011154: human-readable page about the subject at Getty's site
- http://vocab.getty.edu/hier/aat/300011154: human-readable page showing the hierarchical position of the subject
- http://vocab.getty.edu/aat/300011154: semantic URI, content-negotiated using 303 redirect
- http://vocab.getty.edu/aat/300011154.html: HTML page showing RDF triples
- http://vocab.getty.edu/aat/300011154.ttl: Turtle
- http://vocab.getty.edu/aat/300011154.nt: NTriples
- http://vocab.getty.edu/aat/300011154.json: JSON
- http://vocab.getty.edu/aat/300011154.jsonld: JSONLD

Example about an AAT contributor: AATNed
- http://vocab.getty.edu/aat/contrib/10000205: semantic URI, content-negotiated using 303 redirect
- http://vocab.getty.edu/aat/contrib/10000205.html: HTML page showing RDF triples
- http://vocab.getty.edu/aat/contrib/10000205.ttl: Turtle
- http://vocab.getty.edu/aat/contrib/10000205.nt: NTriples
- http://vocab.getty.edu/aat/contrib/10000205.json: JSON
- http://vocab.getty.edu/aat/contrib/10000205.jsonld: JSONLD

Example about an AAT source: Van Nostrand's Scientific Encyclopedia
- http://vocab.getty.edu/aat/source/2000041891: semantic URI, content-negotiated using 303 redirect
- http://vocab.getty.edu/aat/source/2000041891.html: HTML page showing RDF triples
- http://vocab.getty.edu/aat/source/2000041891.ttl: Turtle
- http://vocab.getty.edu/aat/source/2000041891.nt: NTriples
- http://vocab.getty.edu/aat/source/2000041891.json: JSON
- http://vocab.getty.edu/aat/source/2000041891.jsonld: JSONLD

1.8 External Ontologies
Our mapping uses a number of external ontologies (as listed in External Prefixes):
- SKOS, SKOSXL, ISO 25964 for representing thesaurus info;
- DC, DCT for common properties;
- BIBO, FOAF for sources and contributors;
- WGS, Schema for geographic information;
- Bio, Schema for agent information;
- PROV for revision history;
- RDF, RDFS, OWL, XSD for system properties;
- R2RML for implementing the conversion.

Current versions of these external ontologies should be loaded in the semantic repository. We use RDF instead of Turtle versions, to avoid the addition of spurious prefixes to the repository (e.g. skos.ttl defines an empty prefix "":"):
 Getty Vocabularies: LOD  
Semantic Representation  

- SKOS: http://www.w3.org/2004/02/skos/core.rdf
- SKOS-XL: http://www.w3.org/2008/05/skos-xl.rdf
- ISO 25964: iso-thes.rdf obtained with

```
wget --header accept:application/rdf+xml http://purl.org/iso25964/skos-thes#
```

We don't load the other ontologies since they don't make useful inferences for us. In particular, DCT makes a lot of highly irrelevant inferences, e.g. dct:source infers dct:relation and dc:relation. (Feel free to load them in your own repository if you like.)

Brief notes about some of these ontologies follow.

### 1.8.1 DC and DCT

Dublin Core is an often-used ontology for defining basic metadata (e.g. title, creator, created, modified, issued, source, language, etc).

It defines two metadata sets: DC Elements (older, often denoted simply DC and using prefix dc: ) and DC Terms (newer, often denoted DC and using prefix dct: ). Their distinguishing characteristics are:

- DC properties allow any values, literal or URI alike. DCT properties are more strict, and a lot of them require URI.
- DC properties stand alone; DCT properties are often defined as sub-properties of DC (or other DCT properties)

By way of example, the two corresponding properties dc:language and dct:language are defined as if:

```plaintext
dc:language a rdfs:Property .
```

- dc:language can take any value, either literal or URI
- dct:language takes only URIs, infers dc:language, and infers that the URI has class dct:LinguisticSystem.

We use DC/DCT for various common properties, e.g. dc:identifier, dct:source, dct:contributor, dc:created, dct:modified. If both a DC and DCT property fit a purpose, we use the DCT property if the target is a URL.

### 1.8.2 SKOS and SKOS-XL

SKOS is a widely used ontology for representing thesauri. SKOS-XL allows you to represent labels as nodes, and attach additional information to them. We assume the reader has basic knowledge of SKOS. You can consult the following sources:

- SKOS Primer
- SKOS Reference
- SKOS-XL in Primer
- SKOS-XL in Reference

The SKOS standard, as any other standard, is the result of certain tradeoffs and timeline constraints. Therefore various issues and topics that were proposed for consideration did not make it into the final standard. The semantic representation of the GVP thesauri pushes the SKOS envelope in many cases, so it is useful to learn about approaches that go beyond the current SKOS standard.

We found this paper very useful: Key choices in the design of Simple Knowledge Organization System (SKOS), Journal of Web Semantics, May 2013.

See also sections SKOS Inference and SKOS-XL Inference in this document.
1.8.3 ISO 25964

ISO 25964 - the international standard for thesauri and interoperability with other vocabularies is the latest ISO standard on thesauri. The ISO 25964 domain model is shown below:

A lot of it corresponds to SKOS/SKOS-XL, see Correspondence between ISO 25964 and SKOS/SKOS-XL Models. But it has additional constructs not covered by SKOS. In particular, skos:Collection has these limitations:

- you can't put them under a Concept
- you can't say explicitly which are Top Collections in a scheme
- you don't have inverse/transitive versions of skos:member

We use iso:ThesaurusArray, which is a subclass of skos:Collection but can be put under a Concept using iso:superOrdinate, see Sorting with Thesaurus Array.

Despite the risks inherent in early adoption of new technology, Getty willingly undertakes this trail-blazing role, because the ISO ontology allows a faithful representation of GVP data, and in order to promote the adoption and deployment of the standard, which is the way to make technical progress.

We provided implementation experience to the ISO technical committee, and contributed suggestions and fixes to the isothes ontology (first published on 30 Sep 2013 at the public-esw-thes@w3.org mailing list). To the best of our knowledge, the application to AAT is the first industrial use of ISO 25964.
1.8.4 BIBO
The Bibliographic Ontology (BIBO) is used by various library projects, including the British National Bibliography. It is described at http://bibliontology.com, and is kept at GitHub. The definition is available in XML OWL. We use BIBO to:
- Represent Source information: a source is represented as bibo:Document
- We use bibo:DocumentPart if there is location information.

1.8.5 FOAF
The Friend of a Friend ontology (FOAF) is a well-known ontology for people, organizations, contacts, etc. We use FOAF to:
- Represent Contributor information: a contributor is represented as foaf:Agent.
- Link a concept to the thing it represents using foaf:focus (see Concept vs Thing Duality)

1.8.6 PROV
Provenance is information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability or trustworthiness.
PROV is a formalization of Provenance by the Provenance Working Group at W3C (PROVG). It defines a model, corresponding serializations, definitions supporting mapping to other provenance models, and examples how to use PROV.
The PROV family of documents includes the following:
- PROV-OVERVIEW, an overview of the PROV family of documents
- PROV-PRIMER, a primer for the PROV data model
- PROV-DM, the PROV data model
- PROV-O, the PROV ontology, an OWL2 ontology allowing the mapping of the PROV data model to RDF.
- PROV-XML, an XML schema for the PROV data model
- PROV-N, a notation for provenance aimed at human consumption
- PROV-CONSTRAINTS, a set of constraints applying to the PROV data model
- PROV-AQ, mechanisms for accessing and querying provenance
- PROV-DICTIONARY introduces a specific type of collection, consisting of key-entity pairs
- PROV-DC provides a mapping between Dublin Core Terms and PROV-O
- PROV-SEM, a declarative specification in terms of first-order logic of the PROV data model
- PROV-LINKS: introduces a mechanism to link across bundles of provenance info
The PROV-O and PROV-DC ontologies are available as: prov-o.ttl, prov-dc-refinements.ttl.
PROV is rather complex (see the paper The rationale of PROV for discussion about the design decisions in PROV). The PROV-DC mapping illustrates the complexity of PROV best. Below are two examples:
1.8.6.1 dct:modified
A single statement dct:modified is mapped to a network of 6 nodes and 9 statements:

PROV considers that the prov:Modify activity uses an unknown old entity (_:input) and generates an unknown new entity (_:output), both being specializations of the entity under consideration. Furthermore, we need to use a prov:Generation node to be able to use prov:atTime and reflect accurately that the modification is in fact a prov:InstantaneousEvent.

1.8.6.2 dct:creator+dct:created
Two statements dct:creator and dct:created are mapped to a network of 8 nodes and 11 statements:
1.8.7 Geographic Ontologies

There is a plethora of ontologies related to geography out there. Here are some lists:

- **LOV Geography vocabs**: 20 ontologies
- **LOV Geometry vocabs**: 10 ontologies
- **Spatial Ontology Community of Practice (SOCoP), Open Ontology Repository (OOR)**. Unfortunately the site is dead, so these are links from archive.org (2012-06). It includes 40 Ontologies with 2578 terms (counts per ontology). But archive.org doesn't have the ontology pages, so we can use only the names.

We considered the use of the following ontologies for TGN representation:

- **WGS and Schema.org**, as described below
- **GeoNames**: large gazetter of geographic places with coordinate info. See [site, ontology doc]. Defines a comprehensive hierarchy of classes and feature codes, which are comparable to TGN Place Types.
- **DBpedia**: includes a huge amount of information about all kinds of topics. But it has a crowd-sourced ontology definition, which shows some problems. E.g. looking at the class Place, two properties are defined for elevation (altitude and elevation), it is unclear which properties to use for latitude/longitude (if you look at a particular resource e.g. [http://live.dbpedia.org/page/Great_Lakes](http://live.dbpedia.org/page/Great_Lakes), you see it's geo:lat, geo:long, georss:point), and which to use for bounding box.
- **LOCN**: EC Core Location Vocabulary. Recently submitted to W3C for adoption. See [doc & namespace, ontology, Joinup News]. Defines properties to connect to GeoSPARQL geometries.
- **FAO Geopolitical**: the UN Food and Agriculture organization has created a comprehensive dataset about countries, economic groupings, etc. See [ontology, a country profile (US)]
- **CRMGeo**: CIDOC CRM (ISO 21127) is a foundational ontology in the cultural heritage domain. In particular, it will likely be used for representing the Getty [Cultural Objects Name Authority (CONA) thesaurus]. CRMgeo is a recent extension for integrating GeoSPARQL geometries in CRM: see [specification](http://ltv.w3.org/trakr/draft/2012/04/052r4-geo-crm-3.0.0-STABLE-wgs84.html) (version 1.0, created Apr 2013, presented Jun 2013 at CRM-SIG), presentation, ontology. CRMGeo can accommodate complex archaeological scenarios: it considers Spacetime Volumes (i.e. variation of something in time and place, or 4D space); and distinguishes between Phenomenal (defined in relation to some historic/cultural event or thing) and Declarative (defined through some coordinate expressions).
- **NGA/USGS**: the primary source of coordinates in TGN are either of the two large U.S. government databases: United States Geological Survey (USGS) and National Geospatial-Intelligence Agency (NGA) (formerly NIMA). See [Geospatial Semantics and Ontology](http://www.spatialreference.org/documents/national/).

Due to scoping considerations, this initial version of TGN includes only WGS and Schema.org. We welcome your feedback about which other geographic ontologies we should include in the TGN representation, with some justification.

1.8.7.1 W3C WGS Geo Ontology

The W3C Geo Ontology (often called WGS84) is the least common denominator for geographic info, see [document](http://www.w3.org/TR/geo/). Because of its simplicity, it has found wide use. It defines:

- Class wgs: SpatialThing: anything that can have a location
- Properties wgs:lat (latitude), wgs:long (longitude) and wgs:alt (altitude).

1.8.7.2 Schema.org Geographic Features

Schema.org defines common classes and properties to be used on web pages, and is supported by the big 4 search engines: Google, Yahoo, Microsoft, and Yandex. It defines geography-related classes, making a distinction between a place (schema:Place) and its geometry (schema:GeoCoordinates and/or schema:GeoShape). We use the following relevant properties:

- schema:geo to connect a place to its geometry
1.8.8 Agent Ontologies

We’ve considered a number of ontologies for representing ULAN. An important one amongst them is CIDOC CRM (see Geographic Ontologies for a brief description). For this phase we have selected two of these ontologies.

1.8.8.1 Bio Ontology

The Biography ontology includes a number of terms for describing biographical information about people, both living and dead. The approach taken is to describe a person's life as a series of interconnected key events (an approach that is very prominent in CIDOC CRM). We use the bio:Event class and the bio:event property to connect an agent to its events.

1.8.8.2 Schema.org Agent Features

Schema.org includes a number of classes and properties that are useful for describing agents. We've selected it in preference to FOAF, since it has more of what we need for ULAN.

- Class schema:Person and schema:Organization (this could mean an incorporated or non-incorporated group).
- Class schema:Event with schema:location. bio:Event is more specifically designated as a life event, so we use both of these classes, see ULAN Life Events.
- schema:nationality. We twist its use a bit, see ULAN Nationalities.
- Biography places: schema:birthPlace, deathPlace, foundationLocation, dissolutionLocation. (In contrast, we use GVP properties for the dates, see Estimated Dates).

Note: schema:dissolutionLocation is missing from schema.org, so we posted issue/302. The comments in that issue acknowledge this omission and indicate that such property is likely to be included (even though in the far future). So we define it in the GVP ontology, with a rdfs:seeAlso link pointing to the issue.

1.9 GVP Ontology

The GVP ontology includes various classes, properties and individuals (values) used in the mapping. The prefix gvp: stands for "Getty Vocabulary Program". We considered using the more telling prefix getty: but decided against it because other Getty institutions (e.g. the Getty Museum) may start publishing LOD soon. Most of the classes and properties are applicable to AAT, TGN, and ULAN.

The ontology is documented at the “namespace document” http://vocab.getty.edu/ontology and is summarized below. You can also get the ontology in XML/RDF and Turtle, either using content negotiation (see Semantic Resolution), or using the direct links http://vocab.getty.edu/ontology.rdf and http://vocab.getty.edu/ontology.ttl respectively. You can also Explore the Ontology with SPARQL queries.

The context where these classes and properties are used is best seen at Semantic Overview.

- **Subject Types**: gvp:Facet, gvp:Hierarchy, gvp:GuideTerm, gvp:Concept, gvp:ObsoleteSubject. These are implemented as subclasses of skos:Concept, skos:Collection, iso:ThesaurusArray.
  - These are analogous to skos:broaden and friends, but apply to any subject type, and introduce finer distinctions.
  - Note: We decided to remove all struck-out properties for the reasons described in Reduced SKOS Inference.
- Properties gvp:prefLabelGVP, gvp:prefLabelLoC: most often these are "parallel to" (used with) skosxl:prefLabel.
- Other **Subject** properties: gvp:parentString, gvp:parentStringAbbrev.
- **Term Characteristics**: gvp:termKind, gvp:termDisplay, gvp:termType, gvp:termPOS, gvp:termFlag.
• **Sort Order** (applies to Subject, Term, Place Type): gvp:displayOrder
• **Historic Information** properties (apply to Subject, Term, Place Type): gvp:historicFlag, gvp:estStart, gvp:estEnd
• Finally, **Associative Relations** are defined as sub-properties of skos:related (there is a large number of these)

Acknowledgements:
• We created the [ontology documentation (namespace document)](mailto:vocablod@getty.edu) with Parrot. Note that going to the URL of a class/property (e.g. [http://vocab.getty.edu/ontology#aat2000_related_to](http://vocab.getty.edu/ontology#aat2000_related_to)) will jump directly to the definition of that class/property. The documentation also includes embedded RDFa “about” attributes.
• We intend to validate the ontology using [OOPS! (the OntOlogy Pitfall Scanner!)](http://vocab.getty.edu/ontology).

## 2 Semantic Representation

All the information in this section applies equally to all GVP vocabularies. This bears repeating: the basic thesaurus structure of all GVP vocabularies is **the same**. AAT uses only this common structure, while the other vocabularies add some specific data (e.g. see [TGN Specifics](http://vocab.getty.edu/ontology#aat2000 related to)).

This principle of representing various kinds of authority info using a common structure is widely used in large libraries. For example the [Linked Data Service](http://vocab.getty.edu/ontology) of the German National Library (DNB) uses such approach. See the [Ontology of the Integrated Authority File (GND)](http://vocab.getty.edu/ontology) and the document [Linked Data Service of the German National Library: Modelling of bibliographic data](http://vocab.getty.edu/ontology).

### 2.1 Semantic Overview

The diagram below provides an overview of the semantic representation.

• Classes are shown like this: `<gvp:Facet>`. * is a wildcard, e.g. gvp:*Concept stands for gvp:Concept, gvp:PhysPlaceConcept, gvp:AdminPlaceConcept, or gvp:PhysAdminPlaceConcept
• URLs are shown below the classes, where `{x}` indicates a numeric ID. * is a wildcard that stands for aat or tgn.
• (…|…) indicates a choice, e.g. gvp:broader ([Non]Preferred stands for gvp:broaderPreferred or gvp:broaderNonPreferred
The diagram shows pretty much everything but glosses over some details:

- **Subjects** are the main entities of GVP vocabularies. They have various **Subject Types** including **Obsolete Subject** (details are glossed over in the diagram).
  - They are implemented using the standard types skos:Collection, iso:ThesaurusArray, skos:Concept skos:OrderedCollection and rdf:List are also used when a Subject's children are ordered (see **Sorting with Thesaurus Array**)
  - Subjects (except Obsolete) have the following info (exactMatch and Associative Relations apply to Concepts only):
    - Connected to the aat: or tgn: ConceptScheme using skos:inScheme (not shown on the diagram)
    - dc:identifier (see **Identifiers**)
    - gvp:parentString and gvp:parentStringAbbrev
    - gvp:displayOrder (see **Sort Order**)
    - **Historic Information**
    - skos:exactMatch to other thesauri (see **Alignment**)
    - gvp:prefLabelGVP, gvp:prefLabelLoC, skosxl:prefLabel or skosxl:altLabel links to **Terms**: and skos:prefLabel, skos:altLabel as dumbed-down versions of these links (SKOS-XL Inference)
    - skos:scopeNote links to **Scope Notes**
    - dct:source links to **Source** (can be to a LocalSource as shown, or directly to a global Source)
    - dct:contributor links to **Contributor**
  - **Subject Hierarchy** relations come in several varieties: Preferred[NonPreferred and Generic|Partitive|Instantial] (details are glossed over in the diagram).
    - They can be accessed uniformly through **GVP Hierarchical Relations**. Appropriate closures (gvp:broaderExtended, gvp:broaderPreferredExtended, etc) are provided.
    - They are implemented using **Standard Hierarchical Relations** (SKOS and ISO). Transitive closures (e.g. skos:broaderTransitive) are also provided, but you should use the GVP appropriate closures instead. The ISO BTG, BTP, BTI relations (i.e. iso:broader(Generic|Partitive|Instantial)) are provided between skos:Concepts.
  - **Associative Relations** (numbered properties like gvp:aat2000_.* or gvp:tgn3510_.*)) provide lateral relations between subjects.
    - Both Hierarchical and Associative relations may carry **Historic Information** attached to a rdf:Statement
  - **Obsolete Subjects** provide some continuity to clients who have used such subjects in their data. They have little info: only skos:prefLabel, schema:endDate (when it was discontinued), and dct:isReplacedBy (if it was merged to another subject).
  - **Terms** provide multilingual subject labels and carry the following information:
    - dc:identifier (see **Identifiers**)
    - Literals: gvp:term, gvp:qualifier and skosxl:literalForm (being "term (qualifier)", or equal to term if there's no qualifier). All these have the same language tag (@lang)
    - dct:language (Language). Coincidentally, if the language is gvp_lang:<lang>, then the URL of the Term is *_term:<identifier>;<lang>.
    - Enumerated **Term Characteristics**: termDisplay, termFlag, termKind, termPOS and termType
    - gvp:displayOrder (see **Sort Order**)
  - **Historic Information**
    - Links to **Source** (can be to a LocalSource as shown, or directly to a global Source). These links can be plain dct:source; or sub-properties thereof, specifying whether the term is Preferred, NonPreferred or AlternatePreferred for the source
    - Links to **Contributor**, Similar to Source links, these can be plain dct:contributor, or sub-properties thereof
  - **Scope Notes** provide multilingual subject definitions and carry the following information:
    - dc:identifier (see **Identifiers**)

contact us: vocablod@getty.edu
2.2 Subject

Subjects are the main entities (units of thought) in the GVP vocabularies. Subjects (except Obsolete) have the following info, described in the respective sections:

- Subjects are connected to the aat: ConceptScheme using skos:inScheme.
- Subject Hierarchy: threads the subjects using custom (gvp:) properties. Standard skos: and iso: properties are also provided
- Associative Relations: apply to Concepts only
- dc:identifier (see Identifiers)
- gvp:displayOrder (see Sort Order)
- Historic Information about historic applicability
- skos:exactMatch to other thesauri (see Alignment): applies to Concepts only
- skos:scopeNote links to Scope Notes
- dct:source links to Source (can be to Local Sources as shown, or directly to a global Source)
- dct:contributor links to Contributor

Subjects also have these specific properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>range</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>gvp:prefLabelGVP</td>
<td>skosxl:Label</td>
<td>Term preferred by the Getty Vocabulary Program. The language is usually English. Applicable to AAT, ULAN, TGN. Most often used with skosxl:prefLabel</td>
</tr>
<tr>
<td>gvp:prefLabelLoC</td>
<td>skosxl:Label</td>
<td>Term preferred by Library of Congress, thus used for cataloging according to AACR2. Applicable to AAT and ULAN. Most often used with skosxl:prefLabel</td>
</tr>
<tr>
<td>skosxl:prefLabel</td>
<td>skosxl:Label</td>
<td>Preferred term (descriptor)</td>
</tr>
<tr>
<td>skosxl:altLabel</td>
<td>skosxl:Label</td>
<td>Non-preferred term (AlternateDescriptor or UsedForTerm)</td>
</tr>
<tr>
<td>gvp:parentString</td>
<td>literal</td>
<td>Preferred labels of the subject's preferred ancestors, listed bottom up. Useful to show the subject's full context. E.g. for 300226882 &quot;baking dishes&quot; is: &quot;bakeware, &lt;vessels for cooking food&gt;, &lt;containers for cooking food&gt;, &lt;culinary containers&gt;, &lt;containers by function or context&gt;, containers (receptacles), Containers (Hierarchy Name), Furnishings and Equipment (Hierarchy Name), Objects Facet&quot;</td>
</tr>
<tr>
<td>gvp:parentStringAbbrev</td>
<td>literal</td>
<td>Same but skips middle levels for brevity. E.g. for 300226882 &quot;baking dishes&quot; is: &quot;bakeware, &lt;vessels for cooking food&gt;, ... Furnishings and Equipment (Hierarchy Name)&quot;</td>
</tr>
</tbody>
</table>

Subjects also have data properties skos:prefLabel, skos:altLabel as dumbed-down versions of skosxl:prefLabel and skosxl:altLabel (see SKOS-XL Inference).
2.2.1 Subject Types

GVP uses different types of subject to construct the levels of the Subject Hierarchy, and to reflect different concepts in the different vocabularies:

<table>
<thead>
<tr>
<th>Type (Vocab)</th>
<th>Definition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facet (all)</td>
<td>One of the major divisions of a vocabulary</td>
<td>Objects Facet (AAT), World (TGN)</td>
</tr>
<tr>
<td>HierarchyName (AAT)</td>
<td>The top of a hierarchy. Not used for indexing or cataloguing.</td>
<td>Containers (Hierarchy Name)</td>
</tr>
<tr>
<td>GuideTerm (AAT, ULAN)</td>
<td>Place holder to create a level in the hierarchy. Not used for indexing or cataloguing.</td>
<td>&lt;vessels for serving and consuming food&gt;</td>
</tr>
<tr>
<td>Concept (AAT)</td>
<td>Proper concept. Used for indexing and cataloguing.</td>
<td>rhyta</td>
</tr>
<tr>
<td>PhysicalPlaceConcept (TGN)</td>
<td>Physical feature, defined by its physical characteristics on planet Earth, including mountains, rivers, and oceans</td>
<td>Amazon River</td>
</tr>
<tr>
<td>AdministrativePlaceConcept (TGN)</td>
<td>Place defined by administrative boundaries and conditions, including inhabited places, nations, and empires</td>
<td>Burgundy region in France</td>
</tr>
<tr>
<td>PhysAdminPlaceConcept (TGN)</td>
<td>Place that is both administrative and physical. Rarely used</td>
<td>Kiik-Koba</td>
</tr>
<tr>
<td>PersonConcept (ULAN)</td>
<td>A person, be that Artist, Non-artist or Unidentified Named Person, see ULAN Hierarchy and Classes</td>
<td>500115493 Albrecht Dürer, 500048836 &quot;Abraham&quot;</td>
</tr>
<tr>
<td>GroupConcept (ULAN)</td>
<td>A group of people, be that incorporated or un-incorporated</td>
<td>500356337 Albrecht Dürer Workshop</td>
</tr>
<tr>
<td>UnknownPersonConcept (ULAN)</td>
<td>Represents an unknown creator of a given nationality</td>
<td>500125282 Unknown Inca</td>
</tr>
<tr>
<td>ObsoleteSubject (all)</td>
<td>Moved out of the publishable hierarchy, or merged to another. See Obsolete Subject</td>
<td>300375205 &quot;shranks&quot; was merged to 300039264 &quot;schranks&quot; (reflected by dct:isReplacedBy)</td>
</tr>
</tbody>
</table>

We have introduced GVP classes for each type, and arranged them in the following class hierarchy:

```
 skos:Collection ---- iso:ThesaurusArray --- gvp:Hierarchy (AAT)
                        `--- gvp:Facet (AAT, TGN, ULAN)
                                      `--- gvp:GuideTerm (AAT, ULAN)
                                               `--- gvp:ObsoleteSubject (AAT, TGN, ULAN)
                                                        `--- gvp:PhysAdminPlaceConcept (TGN)
                                                             `--- gvp:PhysicalPlaceConcept (TGN)
                                                                  `--- gvp:GroupConcept (ULAN)
                                                                   `--- gvp:PersonConcept (ULAN)
                                           `--- gvp:Subject (AAT)
```

• All are subclasses of gvp:Subject, to allow the user to find all GVP subjects easily. You can restrict by vocabulary by adding e.g. "skos:inScheme aat:" (note the colon at the end!)
• gvp:Facet, gvp:Hierarchy and gvp:GuideTerm are implemented as a subclass of iso:ThesaurusArray (which itself is a subclass of skos:Collection) since they can hold other subjects, but cannot be used for indexing
• gvp:*Concept are implemented as a subclass of skos:Concept, since they are used for indexing
• gvp:ObsoleteConcept is not a subclass of any standard class, since normal thesaurus operations should NOT use nor display them

When a Subject's children are ordered, it is also declared skos:OrderedCollection (see Sorting with Thesaurus Array)

2.3 Subject Hierarchy

The GVP hierarchy includes subjects other than Concepts (see Subject Types); distinguishes between a Preferred parent and optional NonPreferred parents, and makes distinctions between broaderGeneric, broaderPartitive and broaderInstantial.

There is a bunch of hierarchical relation properties, which we explain in detail below, including which properties are derived from which others. Then we explain how the properties are used across the Hierarchy Structure, and which property is used for which subject types.

2.3.1 Standard Hierarchical Relations

SKOS and ISO 25964 provide a number of hierarchical relations. We use the following (d shows the direction up/down):

<table>
<thead>
<tr>
<th>Relation</th>
<th>d</th>
<th>Domain</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>skos:broadeter</td>
<td>▲</td>
<td>skos:Concept</td>
<td>skos:Concept</td>
<td>Parent concept of a concept</td>
</tr>
<tr>
<td>iso:broadeterGeneric</td>
<td>▲</td>
<td>skos:Concept</td>
<td>skos:Concept</td>
<td>Parent in the case of Genus/Species relation</td>
</tr>
<tr>
<td>iso:broadeterPartitive</td>
<td>▲</td>
<td>skos:Concept</td>
<td>skos:Concept</td>
<td>Parent in the case of Part/Whole relation</td>
</tr>
<tr>
<td>iso:broadeterInstantial</td>
<td>▲</td>
<td>skos:Concept</td>
<td>skos:Concept</td>
<td>Parent in the case of Kind/Instance relation</td>
</tr>
<tr>
<td>skos:broadeterTransitive</td>
<td>▲</td>
<td>skos:Concept</td>
<td>skos:Concept</td>
<td>Ancestor concepts (transitive version of broader)</td>
</tr>
<tr>
<td>iso:superOrdinate</td>
<td>▲</td>
<td>iso:ThesaurusArray</td>
<td>skos:Concept</td>
<td>Parent concept of array</td>
</tr>
<tr>
<td>skos:narrower</td>
<td>▼</td>
<td>skos:Concept</td>
<td>skos:Concept</td>
<td>Children concepts of a concept</td>
</tr>
<tr>
<td>skos:narrowerTransitive</td>
<td>▼</td>
<td>skos:Concept</td>
<td>skos:Concept</td>
<td>Descendant concepts (transitive version of narrower)</td>
</tr>
<tr>
<td>skos:member</td>
<td>▼</td>
<td>iso:ThesaurusArray</td>
<td>skos:Concept</td>
<td>Children concepts/arrays of array. See skos:member Structure for an illustration. skos:memberList is also used if the array is ordered, see skos:memberList Structure</td>
</tr>
<tr>
<td>iso:subordinateArray</td>
<td>▼</td>
<td>skos:Concept</td>
<td>iso:ThesaurusArray</td>
<td>Children arrays of a concept</td>
</tr>
</tbody>
</table>

We decided to remove all struck-out properties for the reasons described in Reduced SKOS Inference

2.3.2 GVP Hierarchical Relations

The standard relations have certain limitations:

• Different properties are used for different nodes in the hierarchy (Subject Types), which does not allow you to access the hierarchy uniformly
• iso:broadeterGeneric, iso:broadeterPartitive, iso:broadeterInstantial (commonly known as BTG, BTP, BTI) are important distinctions of the broader relation. These apply to skos:Concept only, but GVP needs to use them for other Subject Types as well.
• These ISO properties are declared sub-properties of skos:broadeter, which itself is a sub-property of skos:broadeterTransitive, so all these relations unconditionally contribute to skos:broadeterTransitive. However, composing these relations is not appropriate in some cases, e.g. Sofia BTP Bulgaria BTI country, but Sofia BTI country is false: Sofia BTI city (or inhabited place), and there is no broader relation between city and country. So skos:broadeterTransitive is not meaningful in these cases.
To overcome these limitations, we introduce a number of custom (GVP) relations, all of which are defined across all subject types (i.e. domain and range is gvp:Subject). The paper On the composition of ISO 25964 hierarchical relations (BTG, BTP, BTI) (V. Alexiev, J. Lindenthal, A. Isaac) discusses the last point (meaningful closure) in detail:

- Introduces "Extended" relations (gvp:broaderGenericExtended, gvp:broaderPartitiveExtended, gvp:broaderInstantialExtended, denoted for brevity BTGE, BTP, BTIE), meant to be meaningful closures of BTG, BTP, BTI.
- Proposes specific compositionality rules to derive the Extended relations as specific chains, see BTG, BTP, BTI Inference.
- Defines gvp:broaderExtended as a disjunction of BTGE, BTP, BTIE.
- Derives the ISO BTG, BTP, BTI relations from those BTGE, BTP, BTIE that connect skos:Concepts directly, see ISO Insert Queries and ISO Rules.

<table>
<thead>
<tr>
<th>Relation</th>
<th>d</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gvp:broader</td>
<td>▲</td>
<td>Parents</td>
<td>Each broader is also Preferred</td>
</tr>
<tr>
<td>gvp:narrower</td>
<td>▼</td>
<td>Children</td>
<td>Inverse of gvp:broader</td>
</tr>
<tr>
<td>gvp:broaderPreferred</td>
<td>▲</td>
<td>Preferred Parent</td>
<td>Main parent, e.g. baking dishes BTG bakeware or Sofia BTP Bulgaria.</td>
</tr>
<tr>
<td>gvp:broaderNonPreferred</td>
<td>▲</td>
<td>Non-PREFERRED</td>
<td>Auxiliary parents, e.g. baking dishes BTG dishes (vessels) and Embrun BTP Alpes Cottiae. There may be several non-preferred parents.</td>
</tr>
<tr>
<td>gvp:broaderGeneric</td>
<td>▲</td>
<td>Parent (Generic)</td>
<td>BTG (Genus/Species, &quot;is a&quot;) relation, e.g. calcite (AAT) BTG mineral (AAT)</td>
</tr>
<tr>
<td>gvp:broaderPartitive</td>
<td>▲</td>
<td>Parent (Partitive)</td>
<td>BTP (Part/Whole, &quot;part of&quot;) relation, e.g. Tuscany (TGN) BTP Italy (TGN)</td>
</tr>
<tr>
<td>gvp:broaderInstantial</td>
<td>▲</td>
<td>Parent (Instantial)</td>
<td>BTI (Kind/Instance, &quot;example of&quot;) relation, eg Rembrandt van Rijn (ULAN) BTI Painter (AAT)</td>
</tr>
<tr>
<td>gvp:broaderGenericExtended</td>
<td>▲</td>
<td>Ancestors (Generic)</td>
<td>Meaningful closure of gvp:broaderGeneric</td>
</tr>
<tr>
<td>gvp:broaderPartitiveExtended</td>
<td>▲</td>
<td>Ancestors (Partitive)</td>
<td>Meaningful closure of gvp:broaderPartitive</td>
</tr>
<tr>
<td>gvp:broaderInstantialExtended</td>
<td>▲</td>
<td>Ancestors (Partitive)</td>
<td>Meaningful closure of gvp:broaderInstantial</td>
</tr>
<tr>
<td>gvp:broaderExtended</td>
<td>▲</td>
<td>Appropriate ancestors</td>
<td>Meaningful closure of gvp:broader for query expansion. Use this, not skos:narrowerTransitive</td>
</tr>
<tr>
<td>gvp:narrowerExtended</td>
<td>▼</td>
<td>Preferred descendants</td>
<td>Meaningful closure of gvp:narrower for query expansion. Use this, not skos:narrowerTransitive</td>
</tr>
<tr>
<td>gvp:broaderPreferredExtended</td>
<td>▲</td>
<td>Preferred Ancestors</td>
<td>Meaningful closure of gvp:broaderPreferred</td>
</tr>
</tbody>
</table>

We decided to remove all struck-out properties for the reasons described in Reduced SKOS Inference.

### 2.3.3 Hierarchy Structure

The structure of the subject hierarchy can be symbolized as F>H>G&C, i.e.

- Facets are above Hierarchies
- Hierarchies are above Guide Terms and Concepts
- Guide Terms and Concepts can be intermixed. There are many examples of G under C. (Note: during vocabulary evolution, G sometimes transitions to C)
The following table shows for each Subj1, what Subj2 can be nested under it (Facet cannot be nested under anything). In each cell we give an example of Subj2, and the linking Standard Hierarchical Relations (e.g. Concept\(\leftarrow\)iso:superOrdinate\(\leftarrow\)GuideTerm).

<table>
<thead>
<tr>
<th>Subj1</th>
<th>Subj2</th>
<th>Hierarchy</th>
<th>Guide Term</th>
<th>Concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facet</td>
<td>Living Organisms skos:member</td>
<td></td>
<td></td>
<td>agents (general) skos:member</td>
</tr>
<tr>
<td>Hierarchy</td>
<td>Visual Works (Hierarchy) skos:member</td>
<td>&lt;organisms by activity&gt; skos:member</td>
<td></td>
<td>visual works skos:member</td>
</tr>
<tr>
<td>Guide Term</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;Early Western World&gt; skos:member</td>
<td>Mediterranean (Early Western World) skos:member</td>
<td></td>
</tr>
<tr>
<td>Concept</td>
<td></td>
<td></td>
<td></td>
<td>Old Hittite Kingdom skos:member</td>
</tr>
</tbody>
</table>

The following diagram illustrates the same nesting possibilities:

You can see that the standard representation is not uniform (for different cases uses different properties and even directions). You may also notice the right-most link skos:broader: although it’s not in the original hierarchy, we insert it, so as to thread the Concepts through the hierarchy. We call this a “thread-through” skos:broader.

Keep in mind that adjacent nodes are also connected uniformly by GVP Hierarchical Relations (gvp:broader going up). Because skos:broader connects only Concepts but "jumps levels" as in the example above, it is neither a sub-property nor a super-property of gvp:broader.

### 2.3.4 Top Concepts

Consider this example from the AAT hierarchy. Legend: S=scheme, F=facet, H=hierarchy, G=guide term, C=concept.

(Some levels are skipped for brevity, inverses and inferred properties are not shown). Every subject in the hierarchy has a link to the AAT ConceptScheme: skos:inScheme aat:
Consider the Concept "vessels (containers)”: it has a thread-through skos:broadern to "containers (receptacles)”. This is a Top Concept, because there are only Facets, Hierarchies and Guide Terms above it. Many SKOS thesauri use skos:topConceptOf (a sub-property of skos:inScheme) to link such Concepts to the ConceptScheme. Even flat concept lists do that, e.g. LoC's MARC Relators are declared topConceptOf http://id.loc.gov/vocabulary/relators (see HTML and RDF).

In a previous version of AAT we did the same, but it has some counter-intuitive consequences. E.g. aat:300054031 "drawing (metalworking)" is a top concept, although it's nested 8 levels deep:

- <metal forming processes and techniques>, <metalworking processes and techniques>, <metalworking processes and techniques>, <processes and techniques by material>, <processes and techniques by specific type>, <processes and techniques>, Processes and Techniques, Activities Facet

The reason is that none of its ancestors is a concept: going up, there are 6 Guide Terms, then a Hierarchy, and finally a Facet. skos:hasTopConcept (the inverse of skos:topConceptOf) is meant as a navigation-enabler, to provide "entry points" in the hierarchy (see SKOS Primer 2.5 Concept Schemes). In AAT the Facets are such entry points, and it is dubious that top concepts would be useful for this purpose. Using skos:hasTopConcept will trick SKOS-consuming applications into displaying a number of disconnected "concept hierarchies", which don't really represent the AAT hierarchy. Picking the roots of "concept hierarchies" at random depths in the general AAT hierarchy feels awkward. There are 887 such hierarchies, the majority of which (633) are no hierarchies at all:

- concepts: 38146
- top concepts: 887 (2.3% of all concepts)
- top concepts without children: 633 (71% of the top concepts)

So we finally decided to omit skos:topConceptOf and skos:hasTopConcept altogether. In lieu of this and to allow RDF crawlers to get all of AAT, we declare all Facets as void:rootResource of the AAT dataset (see Dynamic Descriptive Properties)
2.4 Estimated Dates

The GVP vocabularies include a number of dates in:

- **Historic Information** regarding
  - terms/names (name usage): AAT, TGN, ULAN
  - hierarchical and associative relations (e.g. when a painter was an apprentice of another, when two painters collaborated, when an organization became part of another, when places were allies of each other, activity requires, role creates, etc.): AAT, TGN, ULAN
  - place and agent types (e.g. TGN: inhabited place, ruin, archaeological site; ULAN: artist, painter, landscape architect, engineer, king, art museum, etc.) Place and agent types may also include historical information such as: when a city was incorporated, when a king reigned or a pope ruled.
  - **ULAN Life Events**: when a person was active, when a king ruled, when a painter was documented, etc
  - **ULAN Biographies**: life dates: person birth/death, group founding/dissolution

schema.org has various date properties: startDate, endDate (Event), birthDate, deathDate (Person), foundationDate, dissolutionDate (Organization). However schema.org has no notion of imprecise dates, and only says the dates should be expressed in ISO 8601 date format. But in history and culture, dates are rarely precise.

Given that dating information about people, places, and things is often uncertain or ambiguous, in the Getty vocabularies a "display date" is combined with two estimated numeric values representing the broadest span of years to be used for retrieving this information. To emphasize that these values are not hard dates, but estimations for retrieval purposes, they are represented with these custom properties: gvp:estStart, gvp:estEnd. For more information please refer to the Dates section of the Editorial Guidelines.

For comparison, CIDOC CRM considers that historic periods do not have sharp starts and ends, but fuzzy "ramp-up" and "ramp down" intervals (see How to implement CRM Time in RDF). CRM uses up to 4 dates to describe a historic event. This corresponds to long established methods in scholarship for estimating dates: see Terminus post quem on Wikipedia.

<table>
<thead>
<tr>
<th>CRM property</th>
<th>Meaning</th>
<th>Latin phrase</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>P82a_begin_of_the_begin</td>
<td>started after this moment</td>
<td>terminus post quem</td>
<td>limit after which</td>
</tr>
<tr>
<td>P81a_end_of_the_begin</td>
<td>started before this moment</td>
<td>terminus a quo</td>
<td>limit from which</td>
</tr>
<tr>
<td>P81b_begin_of_the_end</td>
<td>finished after this moment</td>
<td>terminus ad quem</td>
<td>limit to which</td>
</tr>
<tr>
<td>P82b_end_of_the_end</td>
<td>finished before this moment</td>
<td>terminus ante quem</td>
<td>limit before which</td>
</tr>
</tbody>
</table>

The GVP start/end dates correspond to **terminus post quem** and **terminus ante quem** (P82a and P82b or the "outer bounds"). You should use the date fields only for retrieval (range searches) and not for display. You should not interpret biographic GVP start/end dates as precise years.

- When a date is indicated as uncertain in a source, GVP often widens the range. For example, dates "ca" (circa) may be represented by a span widened by 5-10 years (the end-user reads *ca.1590 - ca. 1630*, which could be widened by 10 years earlier and later for retrieval *estStart:1580 estEnd:1640*). Ranges for ancient dates could be widened more broadly, depending upon context and available information.
- When an end date is unknown or estimated as a future date (e.g., a person is still alive), it is often set to start +100 years. As GVP vocabularies are maintained over time, estimated dates would ideally be replaced with actual dates when possible, e.g. based on death notices or other published source updates.

You cannot know whether start/end represent actual or estimated dates. Thus, do not display them to end users. Instead of the GVP estStart/estEnd fields, display the following fields:

- rdfs:comment (Historic info, Events)
- schema:description (Biography), a "one-line biography" of the agent
Most often the display fields include all available date information, plus useful qualifiers (e.g. circa, about, before, after, century, etc). E.g.:

<table>
<thead>
<tr>
<th>field</th>
<th>display string</th>
<th>start</th>
<th>end</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>schema:description</td>
<td>Sri Lankan architect, born 1921</td>
<td>1921</td>
<td>2021</td>
<td>still alive, added 100 years</td>
</tr>
<tr>
<td>schema:description</td>
<td>American art museum, founded 1923</td>
<td>1923</td>
<td>9999</td>
<td>ongoing concern, no real end foreseen</td>
</tr>
<tr>
<td>schema:description</td>
<td>New Kingdom, 18th dynasty (1404-1365 BCE)</td>
<td>-1404</td>
<td>-1365</td>
<td>years BCE represented by negative integers; this isn’t a life span per se, but the period during which the person lived</td>
</tr>
<tr>
<td>schema:description</td>
<td>born after 20 BCE</td>
<td>-20</td>
<td>80</td>
<td>death date unknown, default 100 year lifespan</td>
</tr>
<tr>
<td>schema:description</td>
<td>existed 1378-1485</td>
<td>1378</td>
<td>1485</td>
<td>corporate body existed during a span</td>
</tr>
<tr>
<td>rdfs:comment</td>
<td>1528-ca.1537</td>
<td>1528</td>
<td>1542</td>
<td>start is precise but end is estimated</td>
</tr>
<tr>
<td>rdfs:comment</td>
<td>ca.1330-ca.1380</td>
<td>1325</td>
<td>1385</td>
<td>both start and end dates are uncertain</td>
</tr>
<tr>
<td>rdfs:comment</td>
<td>1210/1212-1246</td>
<td>1210</td>
<td>1246</td>
<td>start is between 1210 and 1212, end is certain</td>
</tr>
<tr>
<td>rdfs:comment</td>
<td>1890s</td>
<td>1890</td>
<td>1899</td>
<td>span is a decade</td>
</tr>
<tr>
<td>rdfs:comment</td>
<td>late 14th century</td>
<td>1375</td>
<td>1399</td>
<td>estimated as last quarter of the century</td>
</tr>
<tr>
<td>rdfs:comment</td>
<td>1516-1527; 1537-1547</td>
<td>1516</td>
<td>1547</td>
<td>complex (non-contiguous) interval; broadest span are the GVP estStart/estEnd dates</td>
</tr>
<tr>
<td>rdfs:comment</td>
<td>ca. 1750</td>
<td>1745</td>
<td>1755</td>
<td>represents a point event: don't assume it lasts 10 years!</td>
</tr>
<tr>
<td>rdfs:comment</td>
<td>année II de la République (1794 CE)</td>
<td>1794</td>
<td>1794</td>
<td>event took place during one year</td>
</tr>
</tbody>
</table>

2.5 Sort Order

Usually, GVP Subjects and Terms are sorted alphabetically (for Subjects, by gvp:prefLabelGVP), but in some cases a specific ("forced") ordering is set, e.g. for many subjects in the Periods and Styles facet. Facets and Hierarchies don't have a useful ordering at present, so we ignore it.

**TGN Place Types** are sorted by a specific Sort Ordering that is meaningful to researchers. According to **TGN Editorial Guidelines**:
- The place type in sequence number 1 must be the Preferred place type.
- The other place types are arranged in reverse chronological order, with Current place types placed before Historical ones.
- Within the subset of current or historical place types that date to the same period, the place types are arranged in order of importance.

To accommodate a maximum number of consumers, we map sorting in 3 ways (belt, suspenders, AND a piece of string!)
1. With a custom property gvp:displayOrder (xsd:positiveInteger). You can use that with "order by" in SPARQL.
2. Ontotext GraphDB preserves the order of nodes as they are **first** inserted in the repository, and returns them in the same order in result sets. We have been careful to load, Subjects, Terms and Place Type relations in the desired order, so you
should also get them in this order, if you don't specify an "order by" clause. Note: there is no guarantee of this behavior, and no W3C standard that mandates it.

3. Using skos:OrderedCollection and iso:ThesaurusArray for Guide Terms and Concepts (see the next section)

2.5.1 Sorting with Thesaurus Array

skos:OrderedCollection defines a standard way to order its children. In addition to skos:member, this uses skos:memberList, being an rdf:List (see section SKOS member vs memberList for details). iso:ThesaurusArray borrows the same paradigm. We illustrate the representation using two examples from the Periods and Styles facet:

- GuideTerm 300106927 <Aegean Bronze Age periods> and Concept 300020224 "Minoan"
- (Note: at present these don't have forced sort order, so instead explore 300018774 <Siberian periods>)

2.5.1.1 skos:member Structure

- The Guide Term (G) is represented as an iso:ThesaurusArray with skosxl:Label "<Aegean Bronze Age periods>"
- The Concept (C) "Minoan" is represented as skos:Concept. But it also has a subordinate array (A), that is an iso:ThesaurusArray, is anonymous, and serves only to hold the skos:OrderedCollection. The anonymous array should not be displayed as a level in the hierarchy.
2.5.1.2 skos:memberList Structure

While the skos:member links establish collection membership (used both with unordered skos:Collection and ordered skos:OrderedCollection), additional skos:memberList structure provides the ordering of the members.

This uses the standard rdf:List construct. People often use blank nodes for rdf:List elements (and there is a special Turtle shortcut for this), but we use explicit URIs since this way it's easier to thread the list with the tooling used (R2RML).

The list URIs have the form `{p}-list-{c}` where `{p}` is the parent ID (owner of the list) and `{c}` is the child ID corresponding to the current list element.

For pedagogical purposes, we first show only the list of (G) "Aegean Bronze Age periods". See the next section for the list of (C) "Minoan" (i.e. the full representation).
2.5.1.3 Full Representation

Adding the skos:memberList of (C) "Minoan", we get the full representation (not for the faint of heart!)
2.6 Associative Relationships

GVP defines a plethora of associative relations between concepts:
- All associative relations are sub-properties of skos:related.
- Relations come in pairs of forward-inverse relation; symmetric relations are self-inverses.
- Every relation instance also has an inverse instance. i.e. if "A rel B" then there is relation "B inv(rel) A".
- We declare pairs as owl:inverseOf (and symmetric relations as owl:SymmetricProperty), but this won't infer new statements.

2.6.1 Relationships Table

The Relationships Table (PDF, XLSX Excel format) describes all associative relationships used by GVP. Legend:
- vocab=which vocabulary it applies to (AAT, TGN, or ULAN)
- fcode=GVP internal code of forward relation, icode=code of inverse relation
- domain=source (of forward relation), range=target (of forward relation); i.e. the kind of Concepts that it can connect
- frel=forward relation name, irl=inverse relation name,
- fdef=forward relation definition, idef=inverse relation definition,
- fexample=example of forward relation, iexample=example of inverse relation.

2.6.2 Relationship Representation

All info from the Relationships Table is represented in RDF.
- Relation URLs are generated as gvp:<vocab><code>_<name> where spaces in the name are replaced with "_".
  - The relation URL includes the vocabulary name and internal GVP code, e.g. aat2208_locus-setting_for (see below). This is a bit unusual for URLs, which are usually written as locusSettingFor. But since some relation names are similar to each other and have undergone some editing, it was felt best to include the code for clarity. At the same time, GVP undertakes to keep relation URLs stable and not change them unless absolutely necessary.
- rdfs:isDefinedBy lists both http://vocab.getty.edu/ontology and the specific vocabulary (e.g. aat: ) so you can find the relations of a specific vocabulary easily.
- All are sub-properties of skos:related. (This was changed recently because there were a few relation instances that connected non-Concepts)
- rdfs:domain and rdfs:range are declared as skos:Concept, whereas the "application" domain & range are emitted as comments only.
- <code> is emitted as dc:identifier 
- ":<name>" - <range>" is emitted as dc:title 
- Examples are emitted as multiple skos:example. We think these add a lot of clarity to the meaning of relations.
- dct:description includes <domain> - <name> - <range> and the examples

```
gvp:aat2208_locus-setting_for a owl:ObjectProperty; 
rdfs:isDefinedBy <http://vocab.getty.edu/ontology>, aat; 
rdfs:subPropertyOf skos:related; 
rdfs:domain skos:Concept; rdfs:range skos:Concept; 
  domain "locus/setting"; range "things"; 
  dc:identifier "2208"; 
skos:prefLabel "aat2208_locus-setting_for"; 
dct:title "locus/setting for - things"; 
skos:example "glassworks (buildings) are the locus/setting for glassware", 
  "caves are the locus/setting for cave paintings"; 
dct:description ""locus/setting - [is] locus/setting for - things. 
  Example: glassworks (buildings) are the locus/setting for glassware; 
  caves are the locus/setting for cave paintings"" .```
TGN and ULAN also define useful associative relations, which use their own prefix (e.g. gvp:aat2811_preceded relates styles/periods/cultures, while gvp:tgn3412_predecessor_of relates nations). For example:

### gvp:tgn3411_successor_of
- **rdfs:isDefinedBy**: <http://vocab.getty.edu/ontology>, tgn: ;
- **rdfs:subPropertyOf** skos:related;
- **rdfs:domain** skos:Concept; **rdfs:range** skos:Concept;
- **dc:identifier** "3411";
- **skos:prefLabel** "tgn3411_successor_of";
- **dc:title** "successor of - nation";
- **skos:example** "Persia is the predecessor of Iran (nation)" ;
- **dct:description** "nation - successor of - nation. Example: Persia is the predecessor of Iran (nation)" .

### gvp:tgn3412_predecessor_of
- **rdfs:isDefinedBy**: <http://vocab.getty.edu/ontology>, tgn: ;
- **rdfs:subPropertyOf** skos:related;
- **rdfs:domain** skos:Concept; **rdfs:range** skos:Concept;
- **dc:identifier** "3412";
- **skos:prefLabel** "tgn3412_predecessor_of";
- **dc:title** "predecessor of - nation" ;
- **dct:description** "nation - predecessor of - nation" .

The dc:title allows you to construct nice displays. E.g. aat:300025419 "rope-makers" has a relation gvp:aat2292_work-live_in to "roperies". So its display can include:

<table>
<thead>
<tr>
<th>Label:</th>
<th>ropemakers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relations:</td>
<td>work/live in - locus/setting</td>
</tr>
</tbody>
</table>

### 2.7 Obsolete Subject

GVP subjects may become obsolete as a result of editorial actions:
- Set as non-publishable (which basically means "deleted")
- Merged to another subject.

Obsolete concepts may have been used in client data. So in order not to leave such data hanging, we publish minimal information about them:
- **skos:prefLabel**: only the GVP preferred label (usually in English)
- **schema:endDate**: when it was obsoleted
- **dct:isReplacedBy**: merged to which subject
Currently AAT obsolete subjects are 4.4% of valid subjects, which shows a good rate of editorial actions, and the importance of this information. Examples from AAT:

```xml
aat:300123456 a gvp:ObsoleteSubject; # Was made non-publishable
    skos:prefLabel "Made up subject";
    skos:inScheme aat:;
    gvp:estEnd "2012-12-31"^^xsd:date.

aat:30038746 a gvp:ObsoleteSubject; # Was merged to a dominant Subject
    skos:prefLabel "Buncheong";
    skos:inScheme aat:;
    dct:isReplacedBy aat:300018699; # Punch'ong
    gvp:estEnd "2012-12-31"^^xsd:date.
```

Merged ObsoleteSubjects are mentioned by ID and name as “Recessive” in the ”merge” Revision History action of the ”Dominant” subject.

We want the inverse relation from the Dominant to the Recessive subject (dct:replaces) as well. DCT doesn't have such declaration, so we add it:

```xml
dct:isReplacedBy owl:inverseOf dct:replaces.
```

Some would say this is "namespace hijacking". We call it adding info that's missing from DCT.

### 2.8 Language

GRI has gathered information about a plethora of Languages (some 1800), both ancient and modern. Other cultural heritage institutions have asked GRI to standardize in this area. Although there are several sources of language information (Lingvo, ISO, etc), a lot of the GRI languages cannot be found there (see GVP Language Tags for details), e.g.:

- Liturgical Greek
- Chinese (transliterated Pinyin without tones)

GVP maintains language data in AAT, as concepts under 300389738 <languages and writing systems by specific example>.

- This data will be used uniformly across all GVP vocabularies
- A mapping to IANA language tags is under development. We have covered all languages used in AAT (about 105) and TGN (about 115 more).
- Unlike AAT where almost all labels have language designators, most of TGN and ULAN do not. This makes sense if you think about it because names of people and places are not really translations. According to the GVP editorial guidelines, language designators can only be added if the source indicates it or if the editor or contributor are experts in a given language. See Importance of the Vernacular Flag.
- All Scope Notes have language.

See the query Languages and ISO Codes to get all language data.

#### 2.8.1 IANA Language Tags

RDF literals use language tags as defined in the IANA Language Subtag Registry. Its structure (described in BCP47 sec 3.1) is not easy to read. So we wrote a script iana-lang-tags.pl that gets the registry, parses it, and writes it to a tab-delimited file. We saved the result as Google Sheet iana-lang-tags (last updated 12 April 2017): search to your heart's content.

The registry includes almost 9000 registrations (broken down by Type and Scope):

- 7769 languages
- 227 extlangs, e.g. ar-auz (Uzbeki Arabic)
- 116 language collections, e.g. bh (Bihari languages)
- 62 macrolanguages, e.g. zh (Chinese), cr (Cree)
- 4 special languages, e.g. und (Undetermined)
- 162 scripts, eg Latn (Latin), Japn (Japanese)
- 301 regions, e.g. US (United States), 021 (Northern America)
- 61 variants
2.8.2 GVP Language Tags

Despite the richness of IANA tags, we had to define new tags, using several extension mechanisms:

- Private language, e.g.
  - x-byzantin-Latin for Byzantine Greek (transliterated)
  - x-khasian for Khasian

- Private language used in specific region, e.g.
  - qqq-002 for African language
  - qqq-142 for Asian language
  - qqq-ET for Ethiopian (not specified which: Boro/Borna, Karo, Male...)

- Private modifier, e.g.
  - grc-Latin-x-liturgie for Liturgical Greek
  - ber-Latin-x-dialect for Berber Dialects (transliterated)
  - fa-Latin-x-middle for Persian, Middle (transliterated)
  - zh-Latin-pinyin-x-notone for Chinese (transliterated Pinyin without tones)
  - x-frisian. IANA/ISO has codes for its predecessor Old Frisian and its dialects West, Saterland and North Frisian, but not for Frisian itself.

GVP has a language 300389645 "Undetermined" (gvp_lang:und). IANA has a corresponding tag und "Undetermined", but we decided NOT to emit it for terms: due to the Open World Assumption, unknown/undetermined info does not need to be emitted. So instead of the info on the left, we emit the one on the right:

<table>
<thead>
<tr>
<th>aat:300312114</th>
<th>aat:300312114</th>
</tr>
</thead>
<tbody>
<tr>
<td>skos:altLabel &quot;bukağı&quot;@und;</td>
<td>skos:altLabel &quot;bukağı&quot;;</td>
</tr>
<tr>
<td>aat_term:1000408720-und</td>
<td>aat_term:1000408720</td>
</tr>
<tr>
<td>skosxl:literalForm &quot;bukağı&quot;@und;</td>
<td>skosxl:literalForm &quot;bukağı&quot;;</td>
</tr>
<tr>
<td>gvp:term &quot;bukağı&quot;@und;</td>
<td>gvp:term &quot;bukağı&quot;.</td>
</tr>
</tbody>
</table>

2.8.3 Language Tag Case

Term URLs have the language tag appended. But you may have noticed a discrepancy in the case used to spell them, e.g.

```
| aat_term:1000024386-en-US xli:literalForm "currycombs"@en-US |
```

This is not an error:

- RFC 5646 (BCP47) section 2.1.1 Formatting of Language Tags demands "At all times, language tags and their subtags, including private use and extensions, are to be treated as case insensitive".
- The SPARQL function `langMatches()` treats them case-insensitively

Nevertheless, it's an unpleasant discrepancy, since RFC 5646 continues "consistent formatting and presentation of language tags will aid users. The format of subtags in the registry is RECOMMENDED as the form to use in language tags." In other words, RFC 5646 demands that implementations are case-insensitive, and recommends that they are case-preserving.

Conventionally, language tags are written in lower-case except:

- Script is capitalized, e.g. el-Latin
- Region is in uppercase, e.g. en-US

You should use langMatches() to compare language tags (see Find Terms by Language Tag), or spell the tag in the exact case used in the repository (e.g. el-latin and en-us).

We undertook some steps to get RDF vendors to normalize case as recommended by RFC 5646, or at least preserve it:
Semantic Representation

- **Posted bug** against Perl's RDF::Trine::Node::Literal (Sep 2013), which is used in the R2RML tool that we use (RDF2RDB)
- Posted **lang_normalize routine** that can be reused by other vendors as well
- Perl **fix adopted and implemented** on Github (Jan 2014)
- Posted bug against Sesame (model and RIO): **SES-1999, SES-1659** (Jan 2014), which is still open
- **Proposed change** to the:
  - RDF 1.1 standard: "Lexical representations of language tags MAY be normalized, according to BCP47 section 2.1.1. "Formatting of Language Tags" (country codes in upper case, script codes capitalized, the rest in lower case). Language tags MAY also be normalized by converting all to lower case, but BCP47 normalization is preferred"
  - SPAROL.lang() function: "lang() MAY normalize the language tag as described in RDF 1.1 Concepts and Abstract Syntax sec 3.3 Literals. It is recommended that lang() normalizes the literal according to BCP47 section 2.1.1, and not by converting it all to lower case."

Jena appears to **store the lang tag as provided**, which is better than storing as lowercase

### 2.8.4 Language Tags and Sources

Language data includes names in several languages, ISO language codes, IANA language tags. E.g. see [300389115](mailto:mailto:300389115@vocablod@getty.edu)

Portuguese:
- Language names (skos:prefLabel): "Portuguese (language)"@en, "portugais"@fr, "Portugiesisch"@de
- Language codes/tags (skos:altLabel): "por"@en, "pt"@en

You can find out where the tags come from by exploring the term's Source:
- The skos:altLabel "pt"@en corresponds to the skosx:altLabel aat_term:1000576998-en that has dct:source aat_source:2000075479, which is "ISO 639-1 Alpha-2 codes for names of languages"
- The skos:altLabel "por"@en corresponds to the skosx:altLabel aat_term:1000576997-en that has dct:source aat_source:2000075493, which is "ISO 639-2 Alpha-3 codes for names of languages"

Notes:
- Previously we had an intermediate bibo:DocumentPart node that was dct:isPartOf aat_source:2000075479 (respectively aat_source:2000075493), with a constant bibo:locator saying "ISO 2-character". We considered this a parasitic node, so we have now removed it
- "pt" and "por" are not English words, so using @en for them is (strictly speaking) not correct. In the future we may introduce special lang tags @x-iso6392, @x-iso6393, @x-iana to mark the language codes with.

Sample queries:
- **Languages and ISO Codes** returns iso2 and iso3 codes
- **Language URLs** returns all GVP languages, and the language tag is the last part of the URL
- **Custom Language Tags** returns the custom tags adopted by GVP

### 2.8.5 Language Dual URLs

AAT languages with assigned language tag have dual URLs, e.g. for Portuguese:
- **aat:300389115**: "systemic" URL of the language as a concept in the Languages hierarchy.
- **gvp_lang:pt**: "logical" URL that includes the language tag is the last part of the URL. These are used as dct:language in Terms (skosx:prefLabel, skosx:altLabel) and Notes (skos:scopeNote). Coincidentally, the URLs of Terms also include the language tag, e.g.

  aat_term:1000011071-pt
  skosx:literalForm "asbestos"@pt;

The dual language URLs are declared owl:sameAs, e.g. **aat:300389115** has this triple:

| aat:300389115 owl:sameAs gvp_lang:pt. |

You can access the language data using either URL. The query **Language URLs** returns all GVP languages.
Why didn't we use established language URLs, e.g. from Lexvo? (For example, the Lingvo site has started using Lexvo URLs in 2010, in a spirit of reuse and cooperation). We would be glad to, but many of the GVP languages are specific so we had to use custom tags. See GVP Language Tags for examples.

2.9 Term

GVP includes multilingual terms and rich information about them, including preferred status, sources, contributors, revision history, etc. We use SKOSXL (skosxl:Label) for the full information, and plain SKOS (literal labels) to allow SKOS-only clients to access the literals (see SKOS and SKOS-XL). Please note that each Term is owned by exactly one Subject (i.e. is dependent on the subject), whereas SKOS-XL potentially allows one xsi:Label to be reused between Concepts (eventually in different roles).

Terms carry the following information:

- **dc:identifier**: numeric ID, also used in the term URL. See Identifiers
- **gvp:term**: the proper (own) label, e.g. "rhea"
- **gvp:qualifier**: serves to clarify and disambiguate terms with the same spelling but different meaning, e.g. "vessels" vs "species" (AAT) or "Republic of", "ancient", "historical" (TGN)
- **skosxl:literalForm**: concatenation of term and parenthesized qualifier, e.g.
  - "rhea (vessels)" meaning "rhyta" (a kind of drinking vessel) vs
  - "rhea (species)" meaning "Boehmeria nivea" (Chinese grass)
- **dct:language** see Language. If the language is gvp_lang:<lang>, then the URL of the Term is aat_term:<identifier>-<lang>, and the language tag of the previous 3 properties is @<lang>
- **gvp:displayOrder**, see Sort Order
- **Historic Information** about historic applicability of the term
- Links to **Source**
  - Can be to a global source (bibo:Document), or to a local source (bibo:DocumentPart)
  - Can be plain dct:source; or sub-properties thereof (gvp:sourcePreferred, gvp:sourceNonPreferred, gvp:sourceAlternatePreferred), specifying whether the term is Preferred, NonPreferred or AlternatePreferred for the source
- Links to **Contributor**.
  - Similar to Source links, these can be plain dct:contributor, or sub-properties thereof (gvp:contributorPreferred, gvp:contributorNonPreferred, gvp:contributorAlternatePreferred)

2.9.1 Term Characteristics

Terms have a number of enumerated characteristics. A lot of these are optional, i.e. could be missing.

- **gvp:termDisplay**: Use for Display (where natural order is preferred) or Indexing/lists (where inverted order is appropriate)?
- **gvp:termFlag**: Vernacular, Loan Term
- **gvp:termKind**: Abbreviation;
  - Common term, Chemical Name, Full term, Jargon or Slang, Neologism, Scientific or Technical term (AAT);
  - FIPS Code, ISO alpha-2 code, ISO alpha-3 code, ISO numeric-3 code, Official Name, Pseudonym, Provisional Name, Site Name, US Postal Service Code (TGN)
- **gvp:termType**: Descriptor, Alternate Descriptor, Used for Term (AAT)

These are captured as Concepts in small subsidiary ConceptSchemes, with their own definitions and examples. You can see these with the query Ontology Values.

Note: in the first release of AAT, the value URLs were under http://vocab.getty.edu/aat. But we have now moved them one level up, because many are also used for TGN.
2.9.2 Importance of the Vernacular Flag
Dealing with languages is a relatively new addition to the TGN, so 80% of TGN labels (1475319 of 1846080) don’t have a language. Still, you can use the following observations:

- Flag Vernacular, which marks a label as using the local language. Most TGN places and ULAN names have at least one Vernacular label.

<table>
<thead>
<tr>
<th>?label gvp:termFlag <a href="http://vocab.getty.edu/term/flag/Vernacular">http://vocab.getty.edu/term/flag/Vernacular</a></th>
</tr>
</thead>
</table>

- Most TGN English labels are marked as such.
- While gvp:prefLabelGVP in AAT is usually in English, in TGN and ULAN it is usually in the Vernacular.

The GVP site for TGN has a feature that allows you to switch between Vernacular Display and English Display, which relies on this. Let's look at a couple of examples from the Greek islands:

- 7011330: Dodekánisos is the prefLabelGVP, which is Vernacular in Greek (transliterated). Sporades (nomos) is the English preferred label, even though it's Historic.
- 7236186: Ágioi Theódoroi, Vrachonisída is the prefLabelGVP, which is Vernacular in Greek (transliterated). There is no English label, so Vernacular Display and English Display show the same. (Island Saint Teodoros is marked with Greek (transliterated), although it is in fact in English)
- 7001338: Iónioi Nísoi is the prefLabelGVP, which is Vernacular, even though it is not marked with a specific language. Ionian Islands is the English preferred label

See [Places, with English or GVP Label](#) for a query that selects the English label if present, else the prefLabelGVP.

2.10 Scope Note
Scope notes define the meaning of a concept, or provide historic description of a place.

- AAT Concepts (but not all non-concept Subjects) have definitions in the basic GVP languages: English, Spanish and Dutch and Chinese (in Progress).
- 2% of TGN records have scope notes, only in English.

Notes are represented with type gvp:ScopeNote and have the following info (compare to Terms):

- dc:identifier: numeric ID, also used in the URL. See [Identifiers](#).
- rdf:value: the note itself (as per [SKOS Primer: 4.2 Advanced Documentation Features](#)), with language tag
- dct:language see [Language](#).
- gvp:displayOrder, see [Sort Order](#)
- Historic Information about applicability
- Links to [Source](#)
- Links to [Contributor](#)

(Note: in the previous release we used type xl:Label and property xl:literalForm to hold the note itself).

2.11 Identifiers
We map the database ID's of Subjects, Terms, Scope Notes, Sources and Contributors to dc:identifier. These are random numeric codes that are also used in the [GVP URLs](#) of these entities.

- They are guaranteed to stay permanent, so you can use them in your own data.
- If a Subject is merged to another, we emit it as [Obsolete Subject](#).

2.12 Notations
A notation is a code or number used to uniquely identify a concept within the scope of a given concept scheme. Unlike Terms, notations are not normally recognizable as a sequence of words in any natural language. DDC, UDC, STW and other well-known thesauri use notations. Example codes include "T58.5" or "303.4833".

We use notations for the following:

- Traditional GVP [Facet/Hierarchy Codes](#), e.g.

| aat:300241490 skos:notation "V.R". # Components (Hierarchy Name) |
• Note: on the GVP website, lower-level subjects (Guide Terms and Concepts) duplicate the same code as the higher level, so we don't include a notation for those

• Short character codes of Term Characteristics, e.g. <http://vocab.getty.edu/term/kind/ScientificOrTechnical> skos:notation "S".

2.13 Source

GVP tracks sources of Terms (labels), Subjects (concepts), and Scope Notes. Sources may be catalogs, encyclopedias, other publications, web sites, databases, etc. AAT, TGN, and ULAN sources are not unified, and so use different prefixes.

We represent sources as bibo:Document with the following info:

• dc:identifier (see Identifiers)
• bibo:shortTitle: brief title
• dct:title: full title
• skos:note: bibliographic note

For example:

aat_source:2000030301 a bibo:Document;
  dc:identifier "2000030301";
  bibo:shortTitle "Chenhall, Revised Nomenclature (1988)";
  dct:title "Chenhall, Robert G. The Revised Nomenclature for Museum Cataloging: ... ".
  aat_source:2000051089 a bibo:Document;
  dc:identifier "2000051089"
  bibo:shortTitle "AATA database (2002-)";
  dct:title "Getty Conservation Institute (GCI). database of AATA Online... 2002-. ".
  aat_source:2000052946 a bibo:Document;
  dc:identifier "2000052946"
  bibo:shortTitle "Encyclopedia Britannica Online (2002-)";
  dct:title "Encyclopædia Britannica. ... http://www.eb.com/ (1 July 2002).".

Sources are applied to Subjects and Scope Notes using dct:source.

For terms, GVP may record whether the term is Preferred, NonPreferred or AlternatePreferred for the source, so we use dct:source or sub-properties thereof: gvp:sourcePreferred, gvp:sourceNonPreferred, gvp:sourceAlternatePreferred.

The multipart URLs like aat_source:2000051089-term-1000198841 are explained in the next section:

# terms of "rhyta" in English, Greek, Spanish
aat_term:1000198841-en
gvp:sourceNonPreferred aat_source:2000049728;

aat_term:1000198841-el-Latn
  gvp:sourceNonPreferred aat_source:2000049728;

aat_term:1000198841-es
  gvp:sourceNonPreferred aat_source:2000049728;

# subject 300198841 (rhyta)
aat:300198841
  dct:source aat_source:2000030301-subject=300198841;
  dct:source aat_source:2000052378.

# notes in English and Dutch
aat_scopeNote:34904
  dct:source aat_source:2000046502.
  aat_scopeNote:83378
  dct:source aat_source:2000051213.
2.13.1 Local Sources

When applied, a source may carry a `bibo:locator` (page number in a document, article name in an encyclopedia, database ID date accessed of a website, etc). To attach this info, we use an intermediate node (Local Source) of type `bibo:DocumentPart` with `dct:isPartOf` that points to the global (original) source.

The URL of local sources consists of the original source URI, followed by the ID of the thing being described (term, subject or note), e.g.:  

```
aat_source:2000051089-term-1000198841 a bibo:DocumentPart;
dct:isPartOf aat_source:2000051089;
bibo:locator "128257 checked 26 January 2012".
```

Note: `bibo:locator` is defined as "A description (often numeric) that locates an item within a containing document or collection", which matches our usage.

Its domain is declared as `bibo:Document`. Consequently our `bibo:DocumentParts` are also inferred to be `bibo:Document`. Although unintended, this is ok, since the two classes are not disjoint (we think this is an omission in BIBO: it should allow `bibo:locator` to "locate a DocumentPart within its containing Document" as well).

2.14 Contributor

GVP tracks contributors of Terms (labels), Subjects (concepts), and Scope Notes. AAT and TGN contributors are not unified, and so use different prefixes.

We represent Contributors as `foaf:Agent` with the following info:

- `dc:identifier` (see Identifiers)
- `foaf:nick`: abbreviation (e.g. CDBP-DIBAM)
- `foaf:name`: full name (e.g. Centro de Documentación de Bienes Patrimoniales, Dirección de Bibliotecas, Archivos y Museos; Santiago, Chile)

For example:

```
aat_contrib:10000000 a foaf:Agent;
dc:identifier "10000000";
foaf:nick "VP";
foaf:name "Getty Vocabulary Program".
aat_contrib:10000131 a foaf:Agent;
dc:identifier "10000131";
foaf:nick "CDBP-DIBAM";
foaf:name "Centro de Documentación de Bienes Patrimoniales ...".
aat_contrib:10000205 a foaf:Agent;
dc:identifier "10000205";
foaf:nick "Bureau AAT";
foaf:name "Bureau AAT, RKD (Netherlands Institute for Art History; ...)".
```

Contributors are attached to Subjects and Notes using `dct:contributor`.

For terms, GVP may record whether the term is Preferred, NonPreferred or AlternatePreferred for the contributor, so we use `dct:contributor` or `sub-properties` thereof: `gvp:contributorPreferred`, `gvp:contributorNonPreferred`, `gvp:contributorAlternatePreferred` (similar to Sources), e.g.:

```
# term "rhyta"
aat_term:1000198841-en
  gvp:contributorNonPreferred aat_contrib:10000131;
  gvp:contributorPreferred aat_contrib:10000000.
  aat_term:1000198841-el-Latn
  gvp:contributorNonPreferred aat_contrib:10000131;
  gvp:contributorPreferred aat_contrib:10000000.
  aat_term:1000198841-es
```

The J. Paul Getty Trust  contact us: vocablod@getty.edu
2.15 Historic Information

GVP includes historic information about the validity of Terms, Hierarchical Relationships, Associative Relationships, TGN Place Types, ULAN Agent Types. We use the following properties:

- **gvp:historicFlag**: with values current, historic, currentAndHistoric
- **gvp:estStart**, **gvp:estEnd**: validity period
  - Years are spelled with least 4 digits. Years BC are expressed as negative, eg: "0100"^^xsd:gYear, "0010"^^xsd:gYear, "-0100"^^xsd:gYear, "-1000"^^xsd:gYear
  - Literals are emitted with proper XSD type according, e.g. "1940"^^xsd:gYear. (In the future there could be dates with higher precision, e.g. "1940-06"^^xsd:gMonthYear or "1940-06-15"^^xsd:date)
  - Previously we used custom sub-properties of dct:valid (defined as "Date (often a range) of validity of a resource"). But many found it confusing to have two values for dct:valid
  - Because the dates are uncertain, we don't use schema:startDate, endDate (see Estimated Dates)
- **rdfs:comment**: note on historic applicability.

2.15.1 Applying to Terms

Applying historic information to Terms represented as explicit nodes (skosxl:Label) is straightforward:

```
aat_term:1000002693-en a skosxl:Label;
skosxl:literalForm "lambruscatura"@en ;
gvp:historicFlag <http://vocab.getty.edu/historic/historic> ;
gvp:estStart "0900"^^xsd:gYear ;
gvp:estEnd "1700"^^xsd:gYear ;
rdfs:comment "Medieval term for wainscoting".
```

2.15.2 Applying to Relations and Types

To apply historic information to relations, we use the RDF Reification vocabulary. It allows to represent a relation instance (i.e. triple) as an rdf:Statement, and use rdf:subject, rdf:object and rdf:predicate to address its components.

We give these statements explicit URLs of the form *-_rel:<subj1>-<type>-<subj2>* where * is the vocabulary, <subj1> and <subj2> are the related subject IDs, and <type> is:

- "broader" for hierarchical relations
- the specific relation name for associative relations (see Relationship Representation)
- "placeType" or "agentType" for TGN place types and ULAN agent types. These also carry gvp:displayOrder (see Sort Order)
AAT examples:

aat_rel:300107346-broader-300020541 a rdf:Statement;
rdf:subject aat:300107346;   # Early Imperial
rdf:predicate gvp:broaderPreferred;
rdf:object aat:300020541;   # Imperial (Roman)
rdfs:comment "ca. 27 BCE-68 CE";
gvp:estStart "-0017"^^xsd:gYear;
gvp:estEnd "0068"^^xsd:gYear .

aat_rel:300020271-aat2812_followed-300020269 a rdf:Statement;
rdf:subject aat:300020271;   # Second Dynasty (Egyptian)
rdf:predicate gvp:aat2812_followed;
rdf:object aat:300020269;   # First Dynasty (Egyptian)
rdfs:comment "Second Dynasty began ca. 2775 BCE";
gvp:estStart "-2785"^^xsd:gYear;
gvp:estEnd "-2765"^^xsd:gYear.

ULAN examples:

ulan_rel:500125816-broader-500125805 a rdf:Statement;
rdf:subject ulan:500125816;   # Museum of Photography
rdf:predicate gvp:broaderPreferred;
rdf:object ulan:500125805;   # Art Library
rdfs:comment "part of Art Library since 2004";
gvp:estStart "2004"^^xsd:gYear.

ulan_rel:500005411-ulan1105_apprentice_of-500003947 a rdf:Statement;
rdf:subject ulan:500005411;   # Genga, Girolamo
rdf:predicate gvp:ulan1105_apprentice_of;
rdf:object ulan:500003947;   # Signorelli, Luca
rdfs:comment "ca. 1494";
gvp:estStart "1493"^^xsd:gYear;
gvp:estEnd "1498"^^xsd:gYear.

You may notice in the last example that start/end do not match rdfs:comment (display date). A lot more about this is explained in Estimated Dates.
2.16 Revision History

GVP keeps extensive info about actions on Subjects and Sources: Although extensive, there is no guarantee the info is comprehensive, especially for very old actions, and for Publish events (issued). The various kinds of actions are listed below (see query Number of AAT Revision Actions for a way to count them). Actions on sub-entities of Subjects (terms, notes, associative relations) are emitted for the Subject.

<table>
<thead>
<tr>
<th>what</th>
<th>dc:type</th>
<th>rdf:type</th>
<th>dc:description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject</td>
<td>created</td>
<td>prov:Create</td>
<td></td>
</tr>
<tr>
<td>Subject</td>
<td>updated</td>
<td>prov:Modify</td>
<td></td>
</tr>
<tr>
<td>Subject</td>
<td>term added</td>
<td>prov:Modify</td>
<td>&lt;term&gt; (&lt;term.id&gt;)</td>
</tr>
<tr>
<td>Subject</td>
<td>term deleted</td>
<td>prov:Modify</td>
<td>&lt;term&gt; (&lt;term.id&gt;) OR was &lt;term&gt;</td>
</tr>
<tr>
<td>Subject</td>
<td>note created</td>
<td>prov:Modify</td>
<td>Language: &lt;lang.name&gt; (&lt;lang.id&gt;)</td>
</tr>
<tr>
<td>Subject</td>
<td>note updated</td>
<td>prov:Modify</td>
<td>Language: &lt;lang.name&gt; (&lt;lang.id&gt;)</td>
</tr>
<tr>
<td>Subject</td>
<td>moved</td>
<td>prov:Modify</td>
<td>Old Parent: &lt;parent.term&gt; (&lt;parent.id&gt;)</td>
</tr>
<tr>
<td>Subject</td>
<td>parent added</td>
<td>prov:Modify</td>
<td>&lt;parent.term&gt; (&lt;parent.id&gt;)</td>
</tr>
<tr>
<td>Subject</td>
<td>relation added</td>
<td>prov:Modify</td>
<td>&lt;this.term&gt; (&lt;this.id&gt;) 'relation.name' &lt;related.term&gt; (&lt;related.id&gt;)</td>
</tr>
<tr>
<td>Subject</td>
<td>relation deleted</td>
<td>prov:Modify</td>
<td>&lt;this.term&gt; (&lt;this.id&gt;) 'relation.name' &lt;related.term&gt; (&lt;related.id&gt;)</td>
</tr>
<tr>
<td>Subject</td>
<td>merged</td>
<td>prov:Modify</td>
<td>Dominant: &lt;this.term&gt; (&lt;this.id&gt;), Recessive: &lt;rec.term&gt; (&lt;rec.id&gt;)</td>
</tr>
<tr>
<td>Subject</td>
<td>issued</td>
<td>prov:Publish</td>
<td>To be published to Website and LOD within 2 weeks</td>
</tr>
<tr>
<td>Source</td>
<td>created</td>
<td>prov:Create</td>
<td></td>
</tr>
<tr>
<td>Source</td>
<td>updated</td>
<td>prov:Modify</td>
<td></td>
</tr>
<tr>
<td>Source</td>
<td>merged</td>
<td>prov:Modify</td>
<td>Dominant: &lt;this.name&gt;, Recessive: &lt;rec.name&gt;</td>
</tr>
</tbody>
</table>

There are over 10M "term updated" actions in AAT alone that are not very interesting and are therefore elided.

2.16.1 Revision History Representation

We map revision actions to PROV-O, using the PROV-DC mapping for inspiration. We use classes and properties from prov-o.ttl and prov-dc-refinements.ttl. But because PROV has a high level of complexity (see the examples in section PROV), we use a rather simplified mapping. Revision records use the *_rev: and *_source_rev: prefixes and have the following data:

- **rdf:type**: prov:Activity, and a specific type as listed in the table above: prov:Create, prov:Modify, or prov:Publish (these come from the PROV-DC mapping)
- **dc:type**: a literal as listed in the table above
- **prov:startedAtTime**: timestamp of the action (xsd:dateTime)
- **dc:description**: additional narrative about the action, such as which Recessive subject was merged, or what is the language of the note that was added.

Links between the entity and its actions:
• skos:changeNote: from entity to action. The SKOS Advanced Documentation pattern shows this for skos:Concept. For uniformity, we use the same property for any gvp:Subject, and also for Sources (foaf:Agents). The property doesn’t define a domain, so that’s permissible.

• prov:wasGeneratedBy: from entity to the prov:Create action
• prov:used: from prov:Modify or prov:Publish to the entity

Timestamps on the entity: we emit the action timestamps also as timestamps on the entity, using DCT properties:

• prov:Create → dct:created
• prov:Modify → dct:modified: there is one for each Modify action. If you prefer to have only the latest timestamp, let us know
• prov:Publish → dct:issued

2.16.2 Revision History for Subject

Let’s say subject 300018699 was created by action 12345, modified by action 12346 (a term was added), and issued (published) by action 12347. We map it thus:

```
aat:300018699
   prov:wasGeneratedBy aat_rev:12345;
   dct:created "2014-01-02T01:02:03"^^xsd:dateTime;
   dct:modified "2014-01-03T01:02:03"^^xsd:dateTime;
   dct:issued "2014-01-04T01:02:03"^^xsd:dateTime.

aat_rev:12345 a prov:Activity, prov:Create;
   dc:type "created";
   prov:startedAtTime "2014-01-02T01:02:03"^^xsd:dateTime.

aat_rev:12346 a prov:Activity, prov:Modify;
   prov:used aat:300018699;
   dc:description "leggings, puttee (1000248060)";
   prov:startedAtTime "2014-01-03T01:02:03"^^xsd:dateTime.

aat_rev:12347 a prov:Activity, prov:Publish;
   prov:used aat:300018699;
   dc:type "issued";
   prov:startedAtTime "2014-01-04T01:02:03"^^xsd:dateTime.
```

2.16.3 Revision History for Source

Let’s say source 2000053040 was created by action 12345 and modified by action 12346 (another source was merged); there are no Publishing actions for sources. (Note: these are different from the Subject actions described above, even if they have the same ID)

```
aat_source:2000053040
   skos:changeNote aat_source_rev:12345, aat_source_rev:12346;
   prov:wasGeneratedBy aat_source_rev:12345;
   dct:created "2014-01-02T01:02:03"^^xsd:dateTime;
   dct:modified "2014-01-03T01:02:03"^^xsd:dateTime.

aat_source_rev:12345 a prov:Activity, prov:Create;
   dc:type "created";
   prov:startedAtTime "2014-01-02T01:02:03"^^xsd:dateTime.

aat_source_rev:12346 a prov:Activity, prov:Modify;
   prov:used aat_source:2000053040;
   prov:startedAtTime "2014-01-03T01:02:03"^^xsd:dateTime.
```
3 Concept vs Thing Duality

Before moving on to TGN and ULAN, we need to discuss a general question about the difference between concepts and denotations. It is a somewhat established practice that a Concept should be treated separately from the Thing that it denotes (Person, Place, cultural Object, monument, historic Event, etc).

- Concepts are the "business objects" (records) of thesaurus management systems, and the daily business of editorial teams like the GVP.
- Things on the other hand exist (or have existed) independently in the real world.
- Events can happen at Places (not Concepts); Events and be enacted by Agents (not Concepts), cultural Objects can be created by Persons/Agents (not Concepts).
- A good "acid test" that helps to understand the difference between concept and denotation is to consider "creation dates". The creation date of a cultural Object or the birth date of a Person is very different from the dct:created field of a Concept, which is the date when it was registered in a particular thesaurus management system.

A good explanation is provided in the blog post Things & their conceptualisations; SKOS, foaf:focus & modelling choices by Pete Johnston, Cambridge University Library, Sep 2011 (search for "Concept Schemes, SKOS and Document Metadata"). This post makes a distinction between three nodes:

- A real-world (non-web) thing: http://dbpedia.org/resource/Napoleonic_Wars: an event taking place between 1800 and 1815, something with a duration in time, which occurred in physical locations, and in which human beings participated
- A concept (skos:Concept): http://id.loc.gov/authorities/subjects/sh85089767: a "conceptualisation of" the Napoleonic Wars, a social and technological artifact "designed to help interconnect", an "abstraction" created by the authors of LCSH for the purposes of classifying works

It is an established practice to use foaf:focus for the denotation link. The 3 URLs are illustrated below, with some typically used types and relations:

You could also read the post What Do URIs Mean Anyway? by Jenni Tennison, UK Open Data Institute, Jul 2011.

3.1 Cons of the Dual Approach

The cons of this approach include:

- More complexity, potential confusion which URL to use.
  In your Cultural Heritage data you should use the Place or Agent URL (tgn:{m}-place, respectively ulan:{m}-agent), and not the concept URL (tgn:{m}, respectively ulan:{m})
- May need to duplicate information about each resource between the dual URLs, e.g.:
  - for TGN: skos:broader and "place is part of" (e.g. crm:P88i_forms_part_of)
  - for ULAN: skos:pref/altLabel and foaf:name
- Harder to co-reference places, people etc across cultural heritage collections, which is an important use case for cross-collection search

3.2 Co-reference and Co-denotation

The usual way to co-reference instances of places, people etc across authority lists or collections is with owl:sameAs. E.g.
By standard OWL semantics, owl:sameAs causes all statements of two URLs (if loaded into the same repository) to be "smushed together". This is required to implement cross-collection search: when the data of several collections use different URLs for Leonardo, we need to make these URLs equivalent somehow. However, one should use skos:exactMatch to say that two Concepts denote the same thing; if one used owl:sameAs, that would lose the individual perspectives of the two editorial teams that created the two Concepts (e.g. will copy dct:created between the two concepts, and cross-graft the two hierarchies together). There is no special semantics associated with skos:exactMatch (it is even weaker than skos:broadMatch), so one doesn't get co-referencing by skos:exactMatch alone.

An in-depth discussion skos:exactMatch vs owl:sameAs discusses the cons of the dual approach and proposes a "co-denotation axiom" at the end. Expanding on this, we could propose the following co-denotation axioms. \(?cN\) are skos:Concept, \(?tN\) are Things, we use N3 Rules notation, and following OWL semantics, \(?t\) owl:sameAs \(?t\) always holds (a thing is equivalent to itself):

- Concepts denoting the same (or equivalent) thing are equivalent:
  \[
  \{\?c1\ foaf:focus \?t1. \?c2\ foaf:focus \?t2. \?t1\ owl:sameAs \?t2\} \implies \{\?c1 skos:exactMatch \?c2\}
  \]

- Equivalent concepts denote the same (or equivalent) thing:
  \[
  \{\?c1 skos:exactMatch \?c2\}. \?c1\ foaf:focus \?t\} \implies \{\?c2\ foaf:focus \?t\}
  \]
  \[
  \{\?c1 skos:exactMatch \?c2\}. \?c1\ foaf:focus \?t1. \?c2\ foaf:focus \?t2\} \implies \{\?t1\ owl:sameAs \?t2\}
  \]

The last will buy us co-reference, but none of these axioms is adopted by the semantic web community (yet).

In the rest of this section we give examples and counter-examples of the Dual approach from well-known libraries and authority lists.

### 3.3 VIAF: pro

The Virtual International Authority File (VIAF) aggregates authority info from about 20 national libraries. It uses the dual practice.

- Page about Leonardo: [http://viaf.org/viaf/24604287](http://viaf.org/viaf/24604287)
- RDF representation (Available from "Record views" at the bottom): [http://viaf.org/viaf/24604287/rdf.xml](http://viaf.org/viaf/24604287/rdf.xml)

The entries from contributing institutions (including ULAN) are represented as skos:Concepts that have a foaf:focus link to the main entity (foaf:Person and rdaGr2:Person):

```
<http://viaf.org/viaf/sourceID/JPG%7C500010879#skos:Concept> a skos:Concept;
  foaf:focus <http://viaf.org/viaf/24604287>.  
```

The foaf:Person has owl:sameAs links equating to URLs in other known sources.

The dual URLs are rendered as <person/> and <person/#skos:Concept>
3.4 FR BnF: pro
The National Library of France (BnF) uses the dual practice. In the middle of the BnF data model is skos:Concept that has foaf:focus to AUTEUR (foaf:Agent) and OEUVRE (frbr-rda:Work):
3.5 UK BL: pro

The British Library (BL) uses the dual practice. The British National Bibliography's Data Model for Books (version 1.4, August 2012) represents agents (persons, families, organizations) in a dual way, distinguishing between Concept and Agent, and links Concept to Agent using foaf:focus:

```xml
<Person-as-Concept BL URI> a blt:PersonConcept; # rdfs:subClassOf skos:Concept
  foaf:focus <Person-as-Agent BL URI>;
  skos:inScheme <id.loc.gov URI for scheme>.
<Person-as-Agent BL URI> a foaf:Agent, dct:Agent, foaf:Person;
  foaf:familyName; foaf:givenName; foaf:name;
  owl:sameAs <VIAF URI if available>.
```

Different properties of Books (or other bibliographic resources) use one or the other of the dual URIs as appropriate:

```xml
<Resource BL URI> a dct:BibliographicResource, bibo:Book;
  dct:subject <Person-as-Concept BL URI>;
  dct:creator <Person-as-Agent BL URI>.
```

3.6 SE KB: pro

The Swedish National Library (KB.SE) uses the dual practice. Consider the RDF for Leonardo: [http://data.libris.kb.se/open/auth/207435.n3](http://data.libris.kb.se/open/auth/207435.n3)
There is also information about creator and subject of various creative works:

- Both Person and Concept are used for dc:subject (I agree that a Person can indeed be the subject of a creative work).
- Not only owl:sameAs to the VIAF person is made, but also skos:exactMatch to the VIAF concept.
- Unfortunately, the owl:sameAs to http://id.loc.gov/authorities/names/n79034525 breaks the scheme. LOC doesn't use the dual practice, so this makes a skos:Person owl:sameAs skos:Concept, which defeats the purpose.

### 3.7 US LoC: cons

The Library of Congress Name Authority File (LCNAF) does **not** use this practice. Consider the RDF for Leonardo: http://id.loc.gov/authorities/names/n79034525.skos.rdf: the URL is declared a skos:Concept, and there is no class for Person.

### 3.8 DE DNB: cons

The German National Library (DNB) does not use this practice. The reason is that the GND ontology **does not use** (is not aligned to) SKOS. Appropriate classes are aligned to FOAF though, e.g.

```
gnd:DifferentiatedPerson rdfs:subClassOf dnb:DistinguishedPerson.
dnb:DistinguishedPerson owl:equivalentClass foaf:Person.
```

Consider the RDF for Leonardo: http://d-nb.info/118640445/about/rdf. There is a single URL of type gnd:DifferentiatedPerson.
4 TGN Specifics

All of the information in Semantic Representation applies equally to AAT, TGN and ULAN. TGN has additional information, which is shown below and described in the following sub-sections.

4.1 TGN Overview

The additional information is attached to the gvp:*PlaceConcept tgn:{c}:

- The TGN concept has TGN Place Types that are AAT concepts. This relation carries gvp:displayOrder (see Sort Order) and Historic Information.
- Following the Concept vs Thing Duality, the PlaceConcept points to a separate node tgn:{c}-place using foaf:focus. It carries the WGS coordinate information, but the Schema coordinate information is in yet another node tgn:{c}-geometry.
- Coordinate Information consists of latitude, longitude, latitude, and (only for regions) bounding box.

4.2 TGN Place Types

TGN includes rich information about Place Types:

- There are almost 1800 place types, ranging from "shantytown" to "undersea mountain chain".
- When applied to a particular place:
  - Place Types are Ordered (gvp:displayOrder)
  - Each place has one Preferred type and may have as many as 10 Non-Preferred types
  - They carry Historic Info: Historic flag, Start, End, Comment (display date), see Applying to Relations and Types
- The average is 2.12 types per place (Jun 2014: 2671142 place type instances for 1259162 places).
E.g. the place types of Machupicchu, Peru are:

- deserted settlement (preferred, current). Start: 1430, End: 1550. Comment: building started ca. 1440; was inhabited until the Spanish conquest of Peru in 1532
- archaeological site (current). Start: 1911. Comment: rediscovered in 1911
- ruins (current)
- inhabited place (historic)
- Inca center (historic). Start: 1440, End: 1550

Types are attached to a place with gvp:placeTypePreferred or gvp:placeTypeNonPreferred. These are sub-properties of gvp:placeType, going from TGN (Place Concept) to AAT (Place Type).

- Place types are maintained in AAT and form BTG hierarchies.
- For example see the hierarchy of aat:300008347 "inhabited places"
- See Places by Direct and Hierarchical Type for a comparison of querying by direct vs hierarchical type.

### 4.3 Coordinate Information

We represent TGN coordinates using two Geographic Ontologies (WGS and Schema.org). For example tgn:3000034 Great Lakes Region:

```
tgn:3000034 a gvp:AdminPlaceConcept; # Great Lakes Region
gvp:broadernPreferred tgn:1000001; gvp:broadernPartitive tgn:1000001; # North and Central America
gvp:tg3000_related_to tgn:7029370; # Great Lakes (lakes)
foaf:focus tgn:3000034-place.
tgn:3000034-place a wgs:SpatialThing, schema:Place;
wgs:lat "45.0000"; wgs:long "-85.0000"; wgs:alt "183.1840";
schema:geo tgn:3000034-geometry.
tgn:3000034-geometry a schema:GeoCoordinates, schema:GeoShape;
schema:latitude 45.0000; schema:longitude -85.0000; schema:elevation 183.1840;
schema:box "-92.0160,43.1560 -92.0160,48.8120 -82.4910,48.8120 -82.4910,43.1560 -92.0160,43.1560".
```

- Like many other geographic ontologies, schema.org makes a difference between a Place and its geometry (GeoCoordinates and GeoShape). So we put the schema:latitude, schema:longitude, schema:elevation properties there
- A few TGN places also have bounding box info. We put it in schema:box as a rectangular polygon of 5 space-separated points (where the first and last coincide), each represented as "long,lat".
- WGS is simpler: it has a class SpatialThing that carries wgs:lat, wgs:long and wgs:alt directly.

### 5 ULAN Specifics

All of the information in Semantic Representation applies equally to AAT, TGN and ULAN. ULAN has additional information, which is shown below and described in the following sub-sections.
### 5.1 ULAN Overview

The additional information is attached to `ulan:{m}` (concept node) and `ulan:{m}-agent` (agent node):

- Following the **Concept vs Thing Duality**, the concept node points to the separate agent node using foaf:focus
- The concept has type `gvp:PersonConcept`, `gvp:GroupConcept` or `gvp:UnknownPersonConcept`. Strangely, `gvp:UnknownPersonConcept` has more the nature of a group than a person, see next section
- The concept has **ULAN Agent Types** that are AAT concepts. This relation may carry `gvp:displayOrder` (see **Sort Order** and **Historic Information**). The concept may have **Associative Relationships** (e.g. parent_of, child_of, master_of, student_of), also with display order and historic information
- The agent has type `schema:Person` or `schema:Organization` according to its facet, see table in next section
- The agent may have **ULAN Nationalities**, **ULAN Biographies**, **ULAN Life Events**

#### 5.2 ULAN Hierarchy and Classes

Compared to AAT, ULAN has a relatively simple hierarchy:

- 5 Facets: described in the rest of this section
- 2 Guide Terms:
  - `ulan:500353455 <named animals>` (a famous race horse)
  - `ulan:500353455 <unknown artists in area of Northern Europe>`
- The rest are individual records (Concept+Agent)
- Some concepts (organizations) have partitive children (sub-organizations)

The 5 facets and the classes they are mapped to are shown below. We also show counts from query **ULAN Facet Counts** as of Feb 2015:

<table>
<thead>
<tr>
<th>ULAN Facet</th>
<th>Concept Class</th>
<th>Agent Class</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>5000000003 Corporate Bodies</td>
<td><code>gvp:GroupConcept</code></td>
<td><code>schema:Organization</code></td>
<td>38848</td>
</tr>
<tr>
<td>500299802 Non-Artists</td>
<td><code>gvp:PersonConcept</code></td>
<td><code>schema:Person</code></td>
<td>6989</td>
</tr>
<tr>
<td>500000002 Persons, Artists</td>
<td><code>gvp:PersonConcept</code></td>
<td><code>schema:Person</code></td>
<td>183895</td>
</tr>
<tr>
<td>500355043 Unidentified Named People</td>
<td><code>gvp:PersonConcept</code></td>
<td><code>schema:Person</code></td>
<td>1903</td>
</tr>
<tr>
<td>500125081 Unknown People by Culture</td>
<td><code>gvp:UnknownPersonConcept</code></td>
<td><code>schema:Organization</code></td>
<td>2217</td>
</tr>
</tbody>
</table>

**Corporate Bodies**: Two or more people working together in a particular place and within a defined period of time; also museums, galleries and most other repositories. Although called "corporate bodies", ULAN has no info to distinguish between incorporated/unincorporated. We map this facet to **schema:Organization**, which also doesn't make such distinction:
Nevertheless, it's very different from Anonymous Masters. E.g.:

- **Non-Artists:** People like donors, patrons, rulers, sitters, art historians, and others whose names are required for indexing art works but who are themselves not artists. We map to `schema:Person`

- **Persons, Artists:** The most populous and important ULAN facet. Includes individuals involved in the creation or production of works of fine art or architecture, for example painters, sculptors, printmakers, architects. Also craftsmen, artisans, engineers, and others who create visual works, even if their works are not considered fine art per se. Also performance artists.

- **Anonymous Masters** with identified oeuvre (style/hand) but whose names are unknown and whose biography is surmised, e.g.
  - Master of the Getty Epistles: One of the four individual hands of the 1520s Hours Workshop of French manuscript illuminators
  - Bern Carnation Masters: a group of Swiss painters who made a series of altar paintings that prominently feature carnations
  - Brunswick Monogrammist: a Flemish painter, possibly identified with Jan van Amstel or Jan Sanders van Hemessen
  - Master of Alkmaar, North Netherlandish painter, active ca. 1490-ca. 1510
  - Master of the Morgan Leaf; or Lippi-Pesellino Follower (but decidedly not Lippi-Pesellino himself)
  - Master of the Aachen Madonna
  - Master of the Dido Panels

- **Unidentified Named People:** people named in original documents, but where the reference is ambiguous and thus no biography or firm identification of the person may be made. The name has usually been found in archival documents. Often the document may mention only a first or last name, thus hampering or prohibiting identification. No scholarly appellation has been attributed and no assessment of their oeuvre has been made (if either of these criteria is met, the record will be an Anonymous Master in the "Persons, Artists" facet). The names of people in this facet are often flagged for "local use": due to the ambiguous nature of their identity, they should be omitted when ULAN is used for indexing or incorporated in a broad retrieval application. E.g.:
  - 500051176 Abinetti; name mentioned in documentation of lot 209 of an auction held 29 Jan 1824 in London.
  - 500047778 Mr. Aelbert; name mentioned in Duijst van Voorhout inventory, Haarlem (1650)
  - 500052533 Mr. Albert; name mentioned in Giallard inventory, 1639

- **Unknown People by Culture:** appellations referring to generic culture or nationality designations that are typically used in cataloging to record unidentified creators with un-established oeuvres (may also be used for anonymous people other than artists.) The appellation for creation in this context refers to the culture in which the work was created, not necessarily to the nationality or culture of the individual artist (who is by definition unknown). Unknown creators are common, especially in certain disciplines, including ancient art, Asian art, African art, aboriginal art, folk art, decorative arts, and Western art dating from the sixteenth century and earlier. This does not refer to one individual, but instead the same subject refers to any of hundreds of anonymous, unidentified artistic personalities, therefore is mapped to schema:Organization. If such record is used on two works of art, there is no claim that the same person made them.

"Anonymous" creators, who according to CCO and CDWA represent one person and have established oeuvres and estimated life dates, are recorded with pseudonyms or other appellations as Anonymous Masters in the "Persons, Artists" facet (see above).

Note this facet has variant label "anonymous" because some repositories use this term to indicate unknown people. Nevertheless, it's very different from Anonymous Masters. E.g.:

- 500125283 unknown Mayan (or simply "Mayan") with nationality "Mayan"
• 500202778 **unknown Abakwariga (Abakwariga cultural designation)** with nationality "Abakwariga"
• 500355202 **unknown Bulgarian (modern)** with nationality "Bulgarian (modern)"

### 5.3 ULAN Agent Types

ULAN includes rich information about Agent Types (also called Roles on the Getty website).

- The total number of types (Mar 2015) is 817, ranging from "abbesses" to "woodworkers" and everything in-between
- Agent types are maintained in AAT and form meaningful BTG hierarchies. For example see the hierarchy of **300025103 artists** (visual artists)
- The average is 1.77 types per agent (total 240343 agents, 425827 type instances)

When applied to a particular agent:

- Place Types are Ordered (gvp:displayOrder)
- Almost every agent has one Preferred type (gvp:agentTypePreferred, total 234465) and may have several Non-Preferred types (gvp:agentTypeNonPreferred total 191362).
- Preferred types are more generic/uniform (e.g. the most populous type "artists (visual artists)" is usually preferred), while non-preferred types are more specific/granular (e.g. "batik artists", "collagists", etc)
- These are sub-properties of gvp:agentType. They go from ULAN (agent concept) to AAT (concept)
- They carry **Historic Info**: Historic flag, Start, End, Comment (display date), see Applying to Relations and Types

Some example types (preferred is first)

- **500115493** Dürer, Albrecht: artist, painter, printmaker, engraver (printmaker), woodcutter, draftsman, illustrator, designer, author, mathematician, theorist, portraitist, religious artist
- **500356337** Albrecht Dürer Workshop: workshop, printmakers, painters
- **500115983** National Gallery of Art: art museum, museum, repository
- **500125283** unknown Mayan: unidentified

Note that above, singular or plural has been used as more appropriate for the particular subject. But in fact "painter" and "painters" are the same concept. This allows you to search for painters (e.g. Dürer) or painter groups (e.g. Dürer Workshop) uniformly. If you want to limit to one, you have to restrict by type

### 5.4 ULAN Nationalities

In ULAN "nationality" is a shorthand for **nationality, culture, race, ethnicity, religion, even sexual orientation**: a significant social grouping or designation of the agent.

- There are 2189 distinct nationalities, e.g. Abakwariga (peoples), Adventism (religion), Afanasievo (culture/style), Aguada (period)
- There are 1.15 nationalities assigned per agent (total 277688 nationality instances)
- Nationalities are represented as AAT concepts and form meaningful BTG hierarchies
- Nationalities often but not always come from hierarchy **aat:300111079 <styles, periods, and cultures by region>**
- Nationalities are not (yet) correlated to TGN places (e.g. countries) nor languages
- Corporate bodies may also have nationalities, e.g. **500115983** National Gallery of Art is American

Nationalities are represented as gvp:nationalityPreferred or gvp:nationalityNonPreferred, which are sub-properties of schema:nationality. Preferred nationalities are more uniform (e.g. German is preferred for Duerer), while non-preferred types may be more specific (e.g. Bavarian is non-preferred for Duerer).

These relations may carry gvp:displayOrder

### 5.5 ULAN Biographies

An agent may have several ULAN biographies. They are represented as class gvp:Biography. One biography (usually by GVP) is connected using gvp:biographyPreferred, the other using gvp:biographyNonPreferred. Each biography may include:

- schema:description: a one-line biography or "display biography". This is an important field suitable to be shown to users
- dct:contributor indicating the contributing institution of this description
• Biography dates: gvp:estStart, estEnd. Use them with caution, see Estimated Dates. Often the dates are estimated by GVP (not necessarily by the contributing institution) from the description
• Biography places (depending on whether the agent is a Person or Organization): schema:birthPlace, deathPlace, foundationLocation, dissolutionLocation (see Schema.org Agent Features about the last property). These are TGN places (tgn:{m}-place)
• schema:gender, which is an AAT concept indicating male (aat:300189559), female (aat:300189557), other, undefined or not-applicable

5.6 ULAN Life Events
An agent may have several life events, represented as follows:
• Represented with both types schema:Event (so they can carry schema:location) and bio:Event (so we can link with bio:event)
• The agent connects to his/her events using gvp:eventPreferred (for one event) or gvp:eventNonPreferred (for other events). These are sub-properties of bio:event
• gvp:displayOrder to sort the events by preference
• Have a dct:type, which is an AAT concept. At present (Mar 2015) there are 45 event types and a total of 75179 events: see query ULAN Events by Type. More will be added in the future.
• May have schema:location, which is a TGN place (tgn:{m}-place)
• rdfs:comment or "display date": this is an important field suitable to be shown to users
• gvp:estStart, estEnd. Use them with caution, see Estimated Dates

6 Additional Features

6.1 Inference
An important decision is what sort of inference level to use in the GVP SPARQL endpoint. Considerations:
• Advantage: the more powerful the inferencing, the fewer facts need to be stated explicitly and the easier to develop and maintain the conversion
• Disadvantage: users that download the files and don't have the same or higher level of inference will be at a disadvantage
We use an approach that uses the advantage while avoiding the disadvantage:
• The conversion process produces only basic facts: Explicit Exports
• Then we use Ontotext GraphDB's powerful inference features (RDFS, OWL RL, OWL QL, custom rules) and INSERT queries to infer all required consequences and materialize them in the repository.
• Then we extract from Ontotext GraphDB export files with all inferred facts: Per-Entity Exports and Total Exports
In this way you can use the Total Exports without need for additional inference, or use the Explicit Exports and provide the inferences described below.
We also run INSERT queries to generate Dynamic Descriptive Properties, but you don't need to do that: you can get the VOID file directly.

6.1.1 Extended Property Constructs
While OWL2 has very powerful class constructs, its property constructs are quite weak. In particular, OWL2 does not support conjunctive properties. For the inferences below, we found it useful to define several extensions. See Extending OWL2 Property Constructs with OWLIM Rules for comparison to OWL2, discussion and more constructs. Notation:
• pN are premises, r is a restriction (just another premise), tN are types, q is the conclusion, t is the axiom holding them together
• p & r is property conjunction (restriction): both properties connect the same nodes
• [t1] p [t2] is type restriction: the source node has type t1 and the target node has type t2 (shown inside the node)
We implement the constructs using [Ontotext GraphDB Rules](#), and provide the implementations for reference below. We would be interested to hear about efficient implementations of these constructs using [SPIN Rules](#) over large datasets (not just in-memory databases).

<table>
<thead>
<tr>
<th>Construct</th>
<th>Illustration</th>
<th>Ontotext GraphDB implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PropChain</td>
<td><img src="#" alt="PropChain Illustration" /></td>
<td>Id: ptop_PropChain</td>
</tr>
<tr>
<td>q &lt;= p1 / p2</td>
<td><img src="#" alt="PropChain Illustration" /></td>
<td>t <a href="">ptop:premise1</a> p1</td>
</tr>
<tr>
<td></td>
<td><img src="#" alt="PropChain Illustration" /></td>
<td>t <a href="">ptop:premise2</a> p2</td>
</tr>
<tr>
<td></td>
<td><img src="#" alt="PropChain Illustration" /></td>
<td>t <a href="">ptop:conclusion</a> q</td>
</tr>
<tr>
<td></td>
<td><img src="#" alt="PropChain Illustration" /></td>
<td>t <a href="">rdf:type</a> <a href="">ptop:PropChain</a></td>
</tr>
<tr>
<td></td>
<td><img src="#" alt="PropChain Illustration" /></td>
<td>x p1 y</td>
</tr>
<tr>
<td></td>
<td><img src="#" alt="PropChain Illustration" /></td>
<td>y p2 z</td>
</tr>
<tr>
<td></td>
<td><img src="#" alt="PropChain Illustration" /></td>
<td>-------------------------------</td>
</tr>
<tr>
<td></td>
<td><img src="#" alt="PropChain Illustration" /></td>
<td>x q z</td>
</tr>
</tbody>
</table>

| PropRestr          | ![PropRestr Illustration](#) | Id: ptop_PropRestr                  |
| q <= p & r         | ![PropRestr Illustration](#) | t <ptop:premise> p                  |
|                    | ![PropRestr Illustration](#) | t <ptop:restriction> r             |
|                    | ![PropRestr Illustration](#) | t <ptop:conclusion> q              |
|                    | ![PropRestr Illustration](#) | t <rdf:type> <ptop:PropRestr>       |
|                    | ![PropRestr Illustration](#) | x p y                              |
|                    | ![PropRestr Illustration](#) | x r y                              |
|                    | ![PropRestr Illustration](#) | -------------------------------    |
|                    | ![PropRestr Illustration](#) | x q y                              |

| PropChainRestr     | ![PropChainRestr Illustration](#) | Id: ptop_PropChainRestr             |
| q <= (p1 / p2) & r | ![PropChainRestr Illustration](#) | t <ptop:premise1> p1                |
|                    | ![PropChainRestr Illustration](#) | t <ptop:premise2> p2                |
|                    | ![PropChainRestr Illustration](#) | t <ptop:restriction> r             |
|                    | ![PropChainRestr Illustration](#) | t <ptop:conclusion> q              |
|                    | ![PropChainRestr Illustration](#) | t <rdf:type> <ptop:PropChainRestr>  |
|                    | ![PropChainRestr Illustration](#) | x p1 y                              |
|                    | ![PropChainRestr Illustration](#) | y p2 z                              |
|                    | ![PropChainRestr Illustration](#) | x r z                              |
|                    | ![PropChainRestr Illustration](#) | -------------------------------    |
|                    | ![PropChainRestr Illustration](#) | x q z                              |

| TypeRestr          | ![TypeRestr Illustration](#) | Id: ptop_TypeRestr                   |
| q <= [t1] p [t2]   | ![TypeRestr Illustration](#) | t <ptop:premise> p                  |
|                    | ![TypeRestr Illustration](#) | t <ptop:type1> t1                   |
|                    | ![TypeRestr Illustration](#) | t <ptop:type2> t2                   |
|                    | ![TypeRestr Illustration](#) | t <ptop:conclusion> q              |
|                    | ![TypeRestr Illustration](#) | t <rdf:type> <ptop:TypeRestr>       |
|                    | ![TypeRestr Illustration](#) | x p y                              |
|                    | ![TypeRestr Illustration](#) | y <rdf:type> t2 [Cut]               |
|                    | ![TypeRestr Illustration](#) | x <rdf:type> t1 [Cut]               |
|                    | ![TypeRestr Illustration](#) | -------------------------------    |
|                    | ![TypeRestr Illustration](#) | x q y                              |
For chains of length 2 (which are most chains seen in practice), PropChain allows a more efficient implementation than owl:propertyChainAxiom, since one doesn't have to unroll the rdf:List of owl:propertyChainAxiom, making intermediate nodes and statements.

It is also more efficient to implement transitivity with PropChain, instead of owl:TransitiveProperty:

- First, owl:TransitiveProperty is a special case of property chain: a self-chain:

  \[?q \text{ a } \text{owl:TransitiveProperty} \iff ?q \text{ owl:propertyChainAxiom } (?q \ ?q)\]

- By itself, the above chain won't infer anything. Thus transitive properties are usually made on the basis of a step (basis) property p:

  \[?p \text{ rdfs:subPropertyOf } ?q.\]

- It is more efficient to use the step property in the chain, instead of making a self-chain; because the reasoner will try to grow the chain only at the end, instead of trying to combine any split of the chain:

  \[?q \text{ owl:propertyChainAxiom } (?q \ ?p).\]

- We include a rule that implements owl:TransitiveProperty as a self-chain:

  \[
  \text{Id: ptop_TransPropAsChain}
  \begin{align*}
  q & \text{ <rdf:type> <owl:TransitiveProperty>} \\
  t & \text{ <rdf:type> <ptop:PropChain>} \\
  t & \text{ <ptop:premise1> q} \\
  t & \text{ <ptop:premise2> q} \\
  t & \text{ <ptop:conclusion> q}
  \end{align*}
  \]

- But for high-volume transitive properties, we can redeclare them with the appropriate non-self-chain to gain a speed advantage.

The [Cut] in TypeRestr means that the inferencer won't try to apply the rule when one of the rdf:type statements is asserted, it will wait until all other premises (in particular "x p y") are asserted.

The constructs are used in specific places below.
6.1.2 Reduced SKOS Inference

The SKOS property hierarchy is shown below, together with an indication of property characteristics:

- **S**=symmetric, **I**=inverse, **T**=transitive, **D**=disjoint:

<table>
<thead>
<tr>
<th>Property</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>skos:semanticRelation</td>
<td></td>
</tr>
<tr>
<td>rdfs:label</td>
<td></td>
</tr>
<tr>
<td>skos:related (S)</td>
<td></td>
</tr>
<tr>
<td>skos:relatedMatch (S)</td>
<td></td>
</tr>
<tr>
<td>skos:broaderTransitive (T)</td>
<td></td>
</tr>
<tr>
<td>skos:broader (I)</td>
<td></td>
</tr>
<tr>
<td>skos:broadMatch (I)</td>
<td></td>
</tr>
<tr>
<td>skos:narrowerTransitive (IT)</td>
<td></td>
</tr>
<tr>
<td>skos:narrower (I)</td>
<td></td>
</tr>
<tr>
<td>skos:mappingRelation</td>
<td></td>
</tr>
<tr>
<td>skos:closeMatch (S)</td>
<td></td>
</tr>
<tr>
<td>skos:exactMatch (ST)</td>
<td></td>
</tr>
<tr>
<td>skos:relatedMatch (S)</td>
<td></td>
</tr>
<tr>
<td>skos:broadMatch (I)</td>
<td></td>
</tr>
<tr>
<td>skos:narrowMatch (I)</td>
<td></td>
</tr>
</tbody>
</table>

The following constructs are used in the SKOS ontology:

- **rdfs:subPropertyOf**, e.g. to infer skos:broader → skos:broaderTransitive
- **owl:TransitiveProperty**: to make the transitive closure, e.g. of skos:broaderTransitive
- **owl:SymmetricProperty**: to infer "A skos:related B" → "B skos:related A"
- **owl:inverseOf**: to infer e.g. skos:broader → skos:narrower and vice versa

However, we found this infers too many triples. E.g. tgn:1000001 North and Central America has 853k descendants. These were repeated as the following properties, which caused the Per-Entity Exports of top places (e.g. World (Facet), North and Central America, etc) to become huge (over 50Mb). So we decided to remove the struck-out properties as superfluous:

- gvp:broaderPartitiveExtended, gvp:broaderPreferredExtended (GVP properties)
- skos:broaderTransitive, iso:broaderPartitive (Standard properties)
- gvp:narrower, gvp:narrowerPartitive (Standard properties)
- iso:narrowerPartitive, skos:narrowerTransitive, iso:subordinateArray (Symmetric properties going down)
- skos:semanticRelation (twice): very unspecific relation

The following diagram shows where we break the inference for SKOS (red ovals):
We run the following SPARQL Update queries after loading the ontologies.

- **Reduce inference for SKOS and ISO**

```sparql
delete where {?x rdfs:subPropertyOf skos:semanticRelation};
delete {?x rdfs:subPropertyOf ?y}
  where {?x owl:inverseOf ?y
```

- **Remove the inference from ISO properties BTG/BTP/BTI to skos:broader, because in ISO Rules we first infer skos:broader, and then ISO BTG/BTP/BTI:**

```sparql
delete data {iso:broaderGeneric rdfs:subPropertyOf skos:broader};
delete data {iso:broaderPartitive rdfs:subPropertyOf skos:broader};
delete data {iso:broaderInstantial rdfs:subPropertyOf skos:broader};
```

You don’t lose querying expressivity, because SPARQL queries can access properties in both directions:

<table>
<thead>
<tr>
<th>instead of this</th>
<th>use that</th>
</tr>
</thead>
</table>
6.1.3 SKOS member vs memberList

The SKOS Reference, see 9 "Concept Collections" has a semantic constraint (S36) that's relevant to Sorting with Thesaurus Array. S36 requires that every item in a skos:memberList should also have a direct link skos:member. The Open Annotation specification, see 4.3 "List" suggests a way to infer the direct links from the list using owl:propertyChainAxiom.

However, we have chosen to omit explicit skos:member links, because we do that for unordered collections anyway, and it would not have been easier to do differently for ordered collections. So this inference rule is not needed.

6.1.4 SKOS-XL Inference

The SKOS recommendation requires every skosx:Label to also be present as a simple SKOS literal label (see requirement S55 Dumbing-Down SKOSXL labels to SKOS lexical labels). This is not formally stated in the SKOS-XL ontology, but we implement it with OWL Property Chains:

```
skos:prefLabel owl:propertyChainAxiom (skosx:prefLabel skosx:literalForm).
skos:altLabel owl:propertyChainAxiom (skosx:altLabel skosx:literalForm).
```

If you don't have an OWL RL compliant repository/reasoner, you can implement the required inference by running the following update queries after loading the Explicit Exports:

```
```

6.1.5 BTG, BTP, BTI Inference

To infer "Extended" versions (meaningful closures) of the GVP Hierarchical Relations, we use the rules defined in the paper On the composition of ISO 25964 hierarchical relations (BTG, BTP, BTI) (V.Alexiev, J.Lindenthal, A.Isaac). We summarize the rules in the following table.

- Each cell expresses a chain of two relations, and whether they can infer a third one
- The row gives the left relation, the column gives the right relation, and the cell gives the conclusion
- BT*x means "BT* or BTxE". In is a reference to the particular inference, used in the next section
- For each positive conclusion, we give an example; and for some negative we give a counter-example

<table>
<thead>
<tr>
<th>left/right</th>
<th>BTGx</th>
<th>BTPx</th>
<th>BTIx</th>
</tr>
</thead>
<tbody>
<tr>
<td>BTGx</td>
<td>12: BTGE: numerous examples</td>
<td>13: BTP: beak irons BTG anvil components BTP &lt;anvils and anvil accessories&gt;</td>
<td>14: no</td>
</tr>
<tr>
<td>BTPx</td>
<td>15: BTPE: anvil components BTP &lt;anvils and anvil accessories&gt; BTG &lt;forging and metal-shaping tools&gt;</td>
<td>16: BTP: Sofia BTP Bulgaria BTP Europe</td>
<td>17: no: Sofia BTP Bulgaria BTI country, but Sofia is no country</td>
</tr>
<tr>
<td>BTIx</td>
<td>18: BTIE: Mt Athos BTI orthodox religious center BTG Christian religious center</td>
<td>19: no, see below</td>
<td>110: no</td>
</tr>
</tbody>
</table>

These inferences can be expressed in 3 equivalent ways:

- Using SPARQL property path notation (as in the paper)
  `left / right -> conclusion`
- Using N3 Rules:
  `{?x left ?y, ?y right ?z} -> {?x conclusion ?y}
- Using OWL property chains (as implemented in the next section)
Regarding I9: Consider the example *Statue of Liberty pedestal* BTI *pedestals* BTP *statues*. The particular *Statue of Liberty pedestal* has no relation to *statues* in general (is neither an instance nor a part of *statues*). In the figure below we could infer the dashed relation (generalization of BTP from one instance thereof), i.e.

![Diagram](image-url)

But in I9 we have only 3 nodes in sequence, not 4 nodes.

### 6.1.6 BTG, BTP, BTI Axioms

For these inferences we cannot use a finite number of INSERTs because the property chains involved are recursive. We need to use rule-based inference. Ontotext GraphDB supports owl:propertyChainAxiom. First we feed BT* to BT*E:

#### # I1: BTG=>BTGE, BTP=>BTPE, BTI=>BTIE: basic inferences

```plaintext
  gvp:broaderGeneric  rdfs:subPropertyOf gvp:broaderGenericExtended.
  gvp:broaderPartitive  rdfs:subPropertyOf gvp:broaderPartitiveExtended.
  gvp:broaderInstantial  rdfs:subPropertyOf gvp:broaderInstantialExtended.
```

These inferences are explained in the previous section:

#### # I2: BTGx/BTGx=>BTGE

```plaintext
  gvp:broaderGenericExtended  owl:propertyChainAxiom
    (gvp:broaderGenericExtended  gvp:broaderGeneric).
```

#### # I3: BTGx/BTPx=>BTPE

```plaintext
  gvp:broaderPartitiveExtended  owl:propertyChainAxiom
    (gvp:broaderGeneric  gvp:broaderPartitiveExtended).
```

#### # I5: BTPx/BTGx=>BTPE

```plaintext
  gvp:broaderPartitiveExtended  owl:propertyChainAxiom
    (gvp:broaderPartitiveExtended  gvp:broaderGeneric).
```

#### # I6: BTPx/BTPx=>BTPE

```plaintext
  gvp:broaderPartitiveExtended  owl:propertyChainAxiom
    (gvp:broaderPartitive  gvp:broaderPartitive).
```

#### # I8: BTIx/BTGx=>BTIE

```plaintext
  gvp:broaderInstantialExtended  owl:propertyChainAxiom
    (gvp:broaderInstantialExtended  gvp:broaderGeneric).
```

This defines gvp:broaderExtended (BTE) as a disjunction of BTGE, BTPE, BTIE:

#### # I11: BTGE|BTPE|BTIE => BTE

```plaintext
  gvp:broaderGenericExtended  rdfs:subPropertyOf gvp:broaderExtended.
  gvp:broaderPartitiveExtended  rdfs:subPropertyOf gvp:broaderExtended.
  gvp:broaderInstantialExtended  rdfs:subPropertyOf gvp:broaderExtended.
```

### 6.1.7 broaderPreferredExtended Rules

We define gvp:broaderPreferredExtended as:

- Meaningful closure (appropriate extension) of gvp:broaderPreferred, or equivalently as
- Specialization of gvp:broaderExtended along broaderPreferred only.
The J. Paul Getty Trust

The first line is a trivial rdfs:subPropertyOf. The second line fits the construct PropChainRestr:

\[
gvp:Infer\_broaderPreferredExtended \Rightarrow gvp:broaderPreferredExtended |
\]
\[
(gvp:broaderPreferredExtended/gvp:broaderPreferred) & gvp:broaderExtended
\]

The second line involves conjunction, so it cannot be implemented with OWL axioms. It also involves recursion, so it cannot be implemented with SPARQL INSERT, unless one is willing to run INSERTS many times, until nothing new is inferred. We would be interested to hear about efficient implementations using SPIN Rules.

You can implement this property with OWL axioms and an insert query, at the expense of introducing an auxiliary property gvp:broaderPreferredTransitive (therefore more triples):

# Axioms

\[
gvp:broaderPreferred\ rdfs:subPropertyOf\ gvp:broaderPreferredTransitive.
gvp:broaderPreferredTransitive\ a\ owl:TransitiveProperty.
\]

# Insert Query

\[
\text{insert}\ \{(?x \ gvp:broaderPreferredTransitive \ ?y) \}
\]

\[
\text{where}\ \{(?x \ gvp:broaderPreferredTransitive \ ?y, \ ?x \ gvp:broaderExtended \ ?y)\}
\]

6.1.8 ISO Insert Queries

We infer SKOS and ISO Standard Hierarchical Relations (skos:member and iso:superOrdinate; implementing Hierarchy Structure and Sorting with Thesaurus Array respectively). Run these queries after loading all Explicit statements:

# Q1: add skos:member below each iso:ThesaurusArray

\[
\text{insert}\ \{(?x \ skos:member \ ?y) \}
\]

\[
\text{where}\ \{(?y \ gvp:broader \ ?x, \ ?x \ a \ iso:ThesaurusArray)\};
\]

# Q2: add iso:superOrdinate from iso:ThesaurusArray to skos:Concept

\[
\text{insert}\ \{(?y \ iso:superOrdinate \ ?x) \}
\]

\[
\text{where}\ \{(?y \ gvp:broader \ ?x, \ ?x \ a \ skos:Concept, \ ?y \ a \ iso:ThesaurusArray)\};
\]

The query below is not used anymore, since we decided not to emit Top Concept indication:

# Q3: add skos:topConceptOf for those Concepts having no broader Concept in the same scheme

\[
\text{insert}\ \{(?x \ skos:topConceptOf \ ?scheme) \}
\]

\[
\text{where}\ \{(?x \ a \ skos:Concept; \ skos:inScheme \ ?scheme) \}
\]

\[
\text{filter\ not\ exists}\ \{(?x \ gvp:broader \ ?y, \ ?y \ a \ skos:Concept; \ skos:inScheme \ ?scheme)\}
\]

6.1.9 ISO Rules

We infer skos:broader and the ISO properties BTG/BTP/BTI as restrictions of GVP BTGE/BTPE/BTIE (the "Extended" relations implemented above), when the "Extended" relation connects two Concepts directly. Thus the SKOS and ISO properties "thread" the skos:Concept hierarchy as a subset of the complete gvp:Subject hierarchy. E.g.:
Unfortunately this cannot be implemented with an "insert where not exists" query, e.g.:

```
# Q4: is wrong because in addition to the indirect path through ?y, there may be a direct path
insert {?x iso:broaderGeneric ?z}
where {?x a skos:Concept. ?z a skos:Concept. ?x gvp:broaderGenericExtended ?z
    filter not exists {
```

There are directly related concepts that are also indirectly related, as you can verify with this query:

```
select *
where {
    ?y gvp:prefLabelGVP [xl:literalForm ?yLab]. filter not exists {?y a skos:Concept}.
    ?z gvp:prefLabelGVP [xl:literalForm ?zLab]. ?z a skos:Concept}
```

E.g. aat:300025276 mosaicists is related directly to aat:300025103 artists (visual artists), but also indirectly through aat:300386777 <artists by medium or work type>.

We first define an auxiliary property `gvp:broaderNonConcept`: a chain of gvp:broader from Concept to GuideTerms, without intervening Concept. Notes:

- We don't care about chains to Hierarchies or Facets, because there are no Concepts above them, see Hierarchy Structure.
- Such chains are not limited to a specific length. E.g. here is a chain Concept-GuideTerm-GuideTerm-Concept: 
  - <plaster, concrete and mortar working equipment> - <equipment by material processed> - equipment.

So `gvp:broaderNonConcept` is defined with recursive rules:

```
gvp:broaderNonConcept <= [gvp:Concept] gvp:broader [gvp:GuideTerm] |
    gvp:broaderNonConcept / gvp:broader [gvp:GuideTerm]
```

This fits the constructs TypeRestr and PropChainType2:

```
gvp:Infer_broaderNonConcept_TypeRestr a ptop:TypeRestr;
    ptop:premise gvp:broader;
    ptop:type1 skos:Concept;
    ptop:type2 gvp:GuideTerm;
    ptop:conclusion gvp:broaderNonConcept.
gvp:Infer_broaderNonConcept_PropChainType2 a ptop:PropChainType2;
    ptop:premise1 gvp:broaderNonConcept;
```
Now we infer `skos:broader` as either connecting two `skos:Concept` directly, or an extension of `gvp:broaderNonConcept`:

```
skos:broader <= [skos:Concept] gvp:broader [skos:Concept] |
gvp:broaderNonConcept / gvp:broader [skos:Concept]
```

This again fits the constructs `TypeRestr` and `PropChainType2`:

```
gvp:Infer_skosBroader_TypeRestr a ptop:TypeRestr;
ptop:premise gvp:broader;
ptop:type1 skos:Concept;
ptop:type2 skos:Concept;
ptop:conclusion skos:broader.
gvp:Infer_skosBroader_PropChainType2 a ptop:PropChainType2;
ptop:premise1 gvp:broaderNonConcept;
ptop:premise2 gvp:broader;
ptop:type2 skos:Concept;
ptop:conclusion skos:broader.
```

Finally, we infer ISO BTG/BTP/BTI as restrictions of GVP BTGE/BTPE/BTIE. We use the fact that `skos:broader` links pairs of directly connected Concepts.

```
gvp:Infer_isoBroaderGeneric a ptop:PropRestr;
ptop:premise gvp:broaderGenericExtended;
ptop:restriction skos:broader;
ptop:conclusion iso:broaderGeneric.
gvp:Infer_isoBroaderPartitive a ptop:PropRestr;
ptop:premise gvp:broaderPartitiveExtended;
ptop:restriction skos:broader;
ptop:conclusion iso:broaderPartitive.
gvp:Infer_isoBroaderInstantial a ptop:PropRestr;
ptop:premise gvp:broaderInstantialExtended;
ptop:restriction skos:broader;
ptop:conclusion iso:broaderInstantial.
```

### 6.1.10 Hierarchical Relations Inference

The following diagram shows the inference paths between hierarchical relations. Legend:

- The numbers refer to sections in this document (e.g. 4.1.5), or specific queries/rules within (e.g. 4.1.8 Q1)
- Standard relations (2.3.1) are shown in blue, GVP relations (2.3.2) in black
- Bold font shows a "closure" relation: transitive or "proper closure" (see 4.1.4). Bold arrow connects the relation being closed (iterated)
- Red arrow shows a restriction relation: the target relation is restricted by the source relation
- "l&r" means that the step property (broaderGeneric) can be to the left or right of the closure (broaderPartitiveExtended); in other property chains the step property can only be to the right
- "Concept-Array" means the target relation is a restriction of the source relation over immediately connected nodes of type skos:Concept and iso:ThesaurusArray. "Array-*" is a restriction from iso:ThesaurusArray to any immediately connected node. "Concept-Concept" is a restriction over two immediately connected skos:Concepts.
- Note: the source relations `gvp:broader( |Non)Preferred` and `gvp:broader(Generic|Partitive|Instantial)` are always used together (a total of 6 combinations), and `gvp:broader` can be inferred from any one of them. The diagram shows inference from `gvp:broader( |Non)Preferred` only, just to keep it tidier.
6.1.11 FTS Insert Queries

Ontotext GraphDB includes a Full-Text Search extension. We use it to provide Full Text Search in the UI (also see Full Text Search Query). To create the two FTS indexes, use the following queries. (luc:setParam is a space-separated list of the full URLs of properties to include in indexing):

```
PREFIX luc: <http://www.ontotext.com/owlim/lucene#>
INSERT DATA {
  luc:includePredicates luc:setParam "http://www.w3.org/2008/05/skos-xl#prefLabel
http://vocab.getty.edu/ontology#term
http://www.w3.org/2008/05/skos-xl#altLabel
http://purl.org/dc/elements/1.1/identifier"
  luc:index luc:setParam "uri"
  luc:moleculeSize luc:setParam "3";
  INSERT DATA { luc:term luc:createIndex "true" };
}
```

```
INSERT DATA {
  luc:includePredicates luc:setParam "http://www.w3.org/2004/02/skos/core#prefLabel
http://www.w3.org/2004/02/skos/core#altLabel http://www.w3.org/2004/02/skos/core#scopeNote
http://www.w3.org/2008/05/skos-xl#literalForm http://www.w3.org/1999/02/22-rdf-syntax-ns#value
http://purl.org/dc/elements/1.1/identifier"
  luc:index luc:setParam "uri"
  luc:moleculeSize luc:setParam "2";
  INSERT DATA { luc:text luc:createIndex "true" };
}
```

If you don’t use Ontotext GraphDB, you’d need to use the repository-specific FTS extensions of the repository that you use.

6.1.12 OntoGeo Insert Query

Ontotext GraphDB provides some Geo-spatial Extensions, see example of their use in TGN-Specific Queries. To enable them, we need to create the OntoGeo geospatial index. This is done with a query like this:

```
PREFIX ontogeo: <http://www.ontotext.com/owlim/geo#>
INSERT DATA { _:b1 ontogeo:createIndex _:b2. }
```

6.2 Alignment

A key potential benefit of LOD is the ability to create and exploit linkages between datasets, e.g. alignments. AAT provides a few alignments, and GVP hopes that external contributors (such as VUA’s Amalgame project, see video) will provide more.
Note: we hope to soon add ULAN alignments to VIAF, US LCNAF, DE GND, JP NDL.

6.2.1 LCSH Alignment

We generate over 400 alignments of AAT to LCSH based on explicit mention of LCSH ID in bibo:locator of a Local Source:

- When the source dct:isPartOf one of the following LCSH renditions:
  - aat_source:2000093511 Library of Congress Authorities online (2002-)
  - aat_source:2000024811 CDMARC Subjects: LCSH (1988-)
- And bibo:locator includes a number of >=8 digits
- OR when bibo:locator consists of a prefix "sh", followed by a number of >=8 digits

This check is performed for sources at both Subject and Term level. The alignment is emitted for the AAT subject.

Take for example **aat:300008736** "waterfalls":

```
aat:300008736 a gvp:Concept;
  gvp:prefLabelGVP aat_term:1000008736-en;
  aat_term:1000008736-en a skosxl:Label
  aat_source:2000093511-term-1000008736 a bibo:DocumentPart;
  dct:isPartOf aat_source:2000046735;
  bibo:locator "sh 85145720" .
  aat_source:2000093511 a bibo:Document;
```

It has a Term (prefLabel) that has a Local Source (bibo:DocumentPart) that is both part of LCSH, and has a bibo:locator matching the "sh" ID pattern. From this we infer the alignment:

```
```

6.2.2 LCNAF Alignment

We generate about 54k alignments of ULAN to LCNAF based on techniques similar to the ones in the previous section.

- We use the following LCNAF sources:
  - ulan_source:2100149014 (LoC Authorities Database)
  - ulan_source:2100042617 (LoC Authorities online)
  - ulan_source:2100153925 (LoC Name Authority File on the Research Libraries Information Network)
- We handle the following LCNAF prefixes: n, nb, no, np, nr, nt, sh.
- For most of them we emit skos:exactMatch.
- The "sh" prefix is special because it indicates a LC subject not a person. Therefore we emit skos:closeMatch, which is weaker property than skos:exactMatch.

For example:

```
ulan:500115493 # ULAN: Albrecht Duerer
  skos:exactMatch <http://id.loc.gov/authorities/names/n79118011>. # LCNAF Abrecht Duerer
ulan:500009666 # ULAN: Pablo Picasso
  skos:closeMatch <http://id.loc.gov/authorities/subjects/sh88000029>. # LCSH Picasso in motion pictures
ulan:500343857 # ULAN: Verene family
  skos:closeMatch <http://id.loc.gov/authorities/subjects/sh2002004503>. # LCSH Verene family
```

Note that there may be some duplicate rows in ULANOut_LOCALignment.nt, because a ULAN subject can be connected to the same LCNAF subject both at the concept, and at the term level.

Note: we hope to soon add ULAN alignments to VIAF, DE GND, JP NDL.

6.2.3 AATNed Alignment

AATNed is the project producing the Dutch translation of AAT. It started publishing LOD earlier than AAT, so some institutions (e.g. the Rijksmuseum) have already started using their URLs (e.g. [http://service.aat-ned.nl/skos/300024521](http://service.aat-ned.nl/skos/300024521)).
AATNed has made the decision to merge into AAT, so the new GVP URLs should be used (e.g. http://vocab.getty.edu/aat/300024521). The IDs correspond, so there is no need to provide alignment links:

\[
\text{dct:isReplacedBy <http://vocab.getty.edu/aat/300024521>}
\]

### 6.3 Forest UI

Forest is a UI framework for creating semantic applications by Ontotext. GVP uses a customized version of Forest that provides the following features (the red numbers on the screen-shots are explained in the text below).

#### 6.3.1 GVP LOD Home Page

The [GVP LOD home page](http://vocab.getty.edu/aat) provides info about the project, links to resources (documentation, queries, Explicit Exports, Total Exports), and updates from [appropriate Twitter tags](http://www.twitter.com) and the [Google support forum](http://groups.google.com) (on the right side).

#### 6.3.2 Querying

- **(1)** SPARQL query endpoint, supporting SPARQL 1.1
  - Accessible both through REST URL, and through an interactive edit box
  - The [SPARQL](http://service.aat-ned.nl/skos/300024521) link always shows a blank edit box, i.e. **forgets your previous query**
  - **(2)** Sample Queries accessible through the [Queries](http://service.aat-ned.nl/skos/300024521) link. This **remembers** your last **edited query** and the last Table of Contents section you selected, and reactivates them
- **(3)** Full Text Search
- **(4)** Table of Contents of the sample queries. The selected section is highlighted and loaded in the bottom frame
- **(5)** Selected section. Please read it carefully: it often describes several query approaches and often the best query is last in the section. When you hover the mouse over a query, it is emphasized and a “SPARQL” button appears. Click the button to copy the query to the edit box.
  - SPARQL edit box with syntax highlighting and auto-indentation
• All External Prefixes and GVP Prefixes used by the representation are predefined in the repository, so you don't need to add them to the query. But if you have a syntax error, the reported line will be off by the number of prefixes you have used, we hope that is a minor nuisance.

• (6) You can specify whether to return only Explicit triples, or also Inferred triples

• (6) You can specify whether to expand results across owl:sameAs and return owl:sameAs assertions. (This applies to Language Dual URLs)

• (7) Splitters that allow you to resize the frames horizontally and vertically

6.3.3 Query Results

• (8) Tooltip over the Queries link showing the last query. Click this link (not the SPARQL link) to edit the last query.

• (9) Number of results: displayed (initially 200) and total.

• Query result pagination: if there are more than 200 results, use the More button at the very bottom.

• (10) Download query results in 9 supported Semantic Formats
6.3.4 Resource View
Forest includes Semantic Resolution of resources (GVP URLs), including content negotiation.

- (11) Resource representations in HTML and 5 Semantic Formats
  - For HTML: if there are more than 200 statements, use the More button at the bottom
  - For the semantic formats: all triples are downloaded (Explicit and Inferred)
  - For the independent entities (Subjects, Sources, Contributors): all triples of all owned objects are also included (see Per-Entity Exports). E.g. a Subject includes all info about its Terms (labels)
- (12) Label of the resource. For subjects this is gvp:prefLabelGVP/xl:literalForm; see next section for other resources
- (13) Tabs to show triples where the resource plays different roles (subject, predicate, object) or all triples
- (14) Selector whether to include Explicit, Inferred or all triples
- (15) Link to the Subject page on GVP’s website
  - Conversely, the GVP website has a back-link to Forest, and links to download each of the semantic formats (11)
- (16) rdfs:seeAlso link: same as (15) but is machine-readable. Compared to (15), it takes an extra click to get to the GVP website: first a page showing the statements about this link (there's only one on the Object tab), then click the Source link
- (17) Link to the GVP page showing the Subject's position in the Hierarchy
- (18) Clickable links to explore all related resources. Hint: if you want to keep the old resource and explore a new resource, use the "Open link in new tab" function of your browser, which can often be invoked with control-click.

6.3.5 Resource Titles
Forest displays smart resource titles depending on the class of the resource. The query used is shown at Smart Resource Title.
If none of the clauses finds an appropriate title, the local part of the URL (after the last slash) is displayed. Examples:

<table>
<thead>
<tr>
<th>Resource</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://vocab.getty.edu/aat/">http://vocab.getty.edu/aat/</a></td>
<td>Art and Architecture Thesaurus</td>
</tr>
<tr>
<td><a href="http://vocab.getty.edu/aat/300131233">http://vocab.getty.edu/aat/300131233</a></td>
<td>archi lanceolati equilateri</td>
</tr>
<tr>
<td><a href="http://vocab.getby.edu/aat/300264092">http://vocab.getby.edu/aat/300264092</a></td>
<td>Objects Facet@en</td>
</tr>
<tr>
<td><a href="http://vocab.getby.edu/aat/300250308-array">http://vocab.getby.edu/aat/300250308-array</a></td>
<td>300250308-array</td>
</tr>
<tr>
<td><a href="http://vocab.getby.edu/aat/300000283-list-300000295">http://vocab.getby.edu/aat/300000283-list-300000295</a></td>
<td>300000283-list-300000295</td>
</tr>
</tbody>
</table>
### 6.4 Full Text Search

You can search for Subjects using the Full Text Search box.

- **Select AAT, TGN or ULAN to limit by thesaurus (the default choice is Any)***

Two indexes are provided that can be selected with a drop-down:

- **Brief**  (luc:term): includes all terms (prefLabels and altLabels) and subject ID (default)
- **Full**  (luc:text): includes all terms, qualifiers, subject ID, and scope notes.

Please let us know if you'd like an index by prefLabels only

This search uses the Lucene FTS engine that is built into Ontotext GraphDB.

The search results include the following columns: subject ID (with link), GVP-preferred term (gvp:prefLabelGVP), abbreviated parent string, abbreviated scope note, and subject type).

- Result pagination is provided
- All language representations are searched, but the results are returned always in English.

The following pre-processing of the query phrase is performed:

- Characters that are not letter/digit/apostrophe are replaced with a space
- Stop words such as “and”, “of”, etc are removed
- Words are wild-carded with *
• A conjunction (AND) is added between the words
• For Chinese hieroglyphs, no analysis is performed and the entered hieroglyphs must match exactly.

For programmatic querying in SPARQL and the exact important details, see Full Text Search Query and the next few queries.

6.5 Descriptive Information

Machine-readable descriptive information is crucial to allow semantic agents to discover, register, crawl, analyze and summarize datasets. It provides the backbone of information for LOD registries such as the DataHub (http://datahub.io). Yet, the creators of the famous LOD cloud diagram (http://lod-cloud.net) report that many datasets are missing basic descriptive and licensing information, which makes it harder for people and agents to consume LOD.

We provide comprehensive info about the AAT dataset, ontology and concept scheme.
• We used this useful list of metadata-description vocabularies from LOV
• We use properties from most of the vocabularies listed in Descriptive Prefixes. Why so many? Because there is overlap between the different vocabularies, yet each has something extra to say
• The basic ontologies are VOID, DCAT (not to be confused with DCT!), ADMS, CC. They are described in the subsections below.
• We also use the ubiquitous DC, DCT; and a few properties from DCTYPE, VANN, VOAG, WDRS, WV.
• The GVP ontology is registered at Linked Open Vocabularies: http://lov.okfn.org/dataset/lov/details/vocabulary_gvp.html
• We hope that our descriptive info fulfills the Guidelines for Collecting Metadata on Linked Datasets in the Data Hub, and intend to validate this with http://validator.lod-cloud.net/
6.5.1 VOID

Vocabulary of Interlinked Datasets (VOID) is the main ontology for describing RDF datasets. The VOID specification Describing Linked Datasets with the VoID Vocabulary was published by W3C on 3 March 2011. The VOID vocabulary definition (namespace document) provides a reference of all classes and properties, and the following domain model:
The slideshare presentation [VoID: Metadata for RDF Datasets](https://www.slideshare.net/richardcyganiak/void-metadata-for-rdf-datasets) (Richard Cyganiak, May 14, 2012, p.13) provides a more lucid domain model:

VOID covers a number of areas:

- Descriptive info (who, when, what) using DC, DCT
- Structural info, interlinking the dataset, its description, data dumps, SPARQL endpoint, used ontology, etc
- Access URLs and mechanisms, e.g. RDF dumps and SPARQL endpoint
- Vocabularies, properties and classes used
- Statistics about size (number of triples), including per property/class
- Resource URI patterns
6.5.2 DCAT
The Data Catalog Vocabulary (DCAT) is a W3C Recommendation published on 16 January 2014. It is used to describe datasets and data catalogs.
6.5.3 ADMS

The Asset Description Metadata Schema (ADMS) is a metadata vocabulary created by the Interoperability Solutions for European Public Administrations (ISA) Programme of the European Commission.

- While DCAT describes data sets, ADMS describes reusable metadata (e.g. xml schemata, generic data models) and reference data (e.g. code lists, taxonomies, dictionaries, vocabularies).
- While DCAT is focused on data catalogs, ADMS is focused on the assets within a catalog

Version 1.00 was released on 18 April 2012. The [ADMS Conceptual Model](#) is based on an earlier ontology called RADION and is fairly complex:

A spreadsheet template is provided, so a user can fill your dataset's metadata, and then Google Refine can create appropriate RDF.

The [EU Open Data Portal](#) uses a metadata vocabulary (EC-ODP) that's quite close to ADMS. For example, see the description of EuroVoc (EU's multilingual thesaurus), and the corresponding RDF file.
6.5.3.1 **W3C ADMS**

After version 1.00, ADMS was contributed to W3C's Government Linked Data Working Group for further development. W3C streamlined the ADMS and converted it to a DCAT profile. The W3C version of ADMS re-uses and subclasses DCAT and other standard vocabularies (e.g. SKOS, DCT) wherever possible and therefore defines a minimal set of classes and properties of its own. The ADMS specification was published on 1 August 2013 as a W3C note.

The domain model was simplified significantly. We work with this streamlined W3C version.
### 6.5.4 Descriptive Entities

We describe the following entities. Luckily, the domain models of VOID, DCAT and ADMS are fairly well aligned, so the entities can be assigned a consistent set of classes. Then the entities are interlinked with Descriptive Properties, more info is attached as Descriptive Properties, and more is computed as Dynamic Descriptive Properties.

<table>
<thead>
<tr>
<th>entity</th>
<th>URL</th>
<th>classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Descriptor</td>
<td><a href="http://vocab.getty.edu/well-known/void">http://vocab.getty.edu/well-known/void</a></td>
<td>void:DatasetDescription, dcat:CatalogRecord</td>
</tr>
<tr>
<td>Datasets (the first one includes the latter ones)</td>
<td><a href="http://vocab.getty.edu/dataset">http://vocab.getty.edu/dataset</a></td>
<td>void:Dataset, dct:Dataset, dcat:Dataset, adms:Asset, cc:Work, dct:Collection</td>
</tr>
<tr>
<td>Home page</td>
<td><a href="http://vocab.getty.edu/">http://vocab.getty.edu/</a></td>
<td>foaf:Document</td>
</tr>
<tr>
<td>GVP ontology</td>
<td><a href="http://vocab.getty.edu/ontology">http://vocab.getty.edu/ontology</a></td>
<td>owl:Ontology</td>
</tr>
<tr>
<td>Documentation</td>
<td><a href="http://vocab.getty.edu/doc">http://vocab.getty.edu/doc</a></td>
<td>foaf:Document</td>
</tr>
<tr>
<td>Vocabularies</td>
<td><a href="http://vocab.getty.edu/aat/">http://vocab.getty.edu/aat/</a></td>
<td>skos:ConceptSchema</td>
</tr>
<tr>
<td></td>
<td><a href="http://vocab.getty.edu/tgn/">http://vocab.getty.edu/tgn/</a></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="http://vocab.getty.edu/ulan/">http://vocab.getty.edu/ulan/</a></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="http://vocab.getty.edu/dataset/tgn/explicit.zip">http://vocab.getty.edu/dataset/tgn/explicit.zip</a></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="http://vocab.getty.edu/dataset/ulan/explicit.zip">http://vocab.getty.edu/dataset/ulan/explicit.zip</a></td>
<td></td>
</tr>
<tr>
<td>Total Exports</td>
<td><a href="http://vocab.getty.edu/dataset/aat/full.zip">http://vocab.getty.edu/dataset/aat/full.zip</a></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="http://vocab.getty.edu/dataset/tgn/full.zip">http://vocab.getty.edu/dataset/tgn/full.zip</a></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="http://vocab.getty.edu/dataset/ulan/full.zip">http://vocab.getty.edu/dataset/ulan/full.zip</a></td>
<td></td>
</tr>
<tr>
<td>Creator/publisher</td>
<td><a href="http://www.getty.edu/research/">http://www.getty.edu/research/</a></td>
<td>foaf:Organization, foaf:Agent</td>
</tr>
<tr>
<td>SPARQL endpoint</td>
<td><a href="http://vocab.getty.edu/sparql">http://vocab.getty.edu/sparql</a></td>
<td></td>
</tr>
<tr>
<td>License</td>
<td><a href="http://opendatacommons.org/licenses/by/1.0/">http://opendatacommons.org/licenses/by/1.0/</a></td>
<td>cc:License, dct:LicenseDocument</td>
</tr>
</tbody>
</table>

A diagram of the entities and their relations follows:
6.5.5 Descriptive Relations

The main descriptive entities are linked with the following relations. Everywhere below where AAT appears, similar statements are made for TGN and ULAN.

<table>
<thead>
<tr>
<th>subjects</th>
<th>relations</th>
<th>objects</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://vocab.getty.edu/well-known/void">http://vocab.getty.edu/well-known/void</a></td>
<td>foaf:primaryTopic</td>
<td><a href="http://vocab.getty.edu/dataset">http://vocab.getty.edu/dataset</a></td>
</tr>
<tr>
<td><a href="http://vocab.getty.edu/dataset">http://vocab.getty.edu/dataset</a></td>
<td>void:subset</td>
<td><a href="http://vocab.getty.edu/dataset/aat">http://vocab.getty.edu/dataset/aat</a></td>
</tr>
<tr>
<td><a href="http://vocab.getty.edu/dataset">http://vocab.getty.edu/dataset</a></td>
<td></td>
<td><a href="http://vocab.getty.edu/dataset/tgn">http://vocab.getty.edu/dataset/tgn</a></td>
</tr>
<tr>
<td><a href="http://vocab.getty.edu/dataset">http://vocab.getty.edu/dataset</a></td>
<td></td>
<td><a href="http://vocab.getty.edu/dataset/ulan">http://vocab.getty.edu/dataset/ulan</a></td>
</tr>
<tr>
<td><a href="http://vocab.getty.edu/dataset">http://vocab.getty.edu/dataset</a></td>
<td>wdrs:describedby</td>
<td><a href="http://vocab.getty.edu/doc">http://vocab.getty.edu/doc</a></td>
</tr>
<tr>
<td><a href="http://vocab.getty.edu/dataset">http://vocab.getty.edu/dataset</a></td>
<td>void:vocabulary</td>
<td><a href="http://vocab.getty.edu/ontology">http://vocab.getty.edu/ontology</a> 1</td>
</tr>
<tr>
<td><a href="http://vocab.getty.edu/dataset/aat">http://vocab.getty.edu/dataset/aat</a></td>
<td>void:rootResource</td>
<td><a href="http://vocab.getty.edu/aat/">http://vocab.getty.edu/aat/</a> (same for TGN, ULAN) 2</td>
</tr>
<tr>
<td><a href="http://vocab.getty.edu/dataset/aat">http://vocab.getty.edu/dataset/aat</a></td>
<td></td>
<td><a href="http://vocab.getty.edu/dataset/aat/full.zip">http://vocab.getty.edu/dataset/aat/full.zip</a></td>
</tr>
<tr>
<td><a href="http://vocab.getty.edu/dataset/aat">http://vocab.getty.edu/dataset/aat</a></td>
<td></td>
<td>(same for TGN, ULAN)</td>
</tr>
<tr>
<td><a href="http://vocab.getty.edu/dataset/aat/explicit.zip">http://vocab.getty.edu/dataset/aat/explicit.zip</a></td>
<td>dcat:accessURL</td>
<td><a href="http://vocab.getty.edu/sparql">http://vocab.getty.edu/sparql</a></td>
</tr>
<tr>
<td><a href="http://vocab.getty.edu/dataset/aat/explicit.zip">http://vocab.getty.edu/dataset/aat/explicit.zip</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="http://vocab.getty.edu/dataset/aat/explicit.zip">http://vocab.getty.edu/dataset/aat/explicit.zip</a></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6.5.6 Descriptive Properties

We use the following descriptive properties and values.

- We use appropriate ADMS SKOS concepts from namespace http://purl.org/adms/ (Turtle). Their URLs are self-describing.
- The distribution of properties amongst entities is dictated by the domain models above (and see the actual VOID descriptor)
- Below we show only AAT; similar statements are made for TGN, ULAN

---

1 And the other External Ontologies used by GVP
2 And all gvp:Facets, see next section
3 Using the void:uriRegexPatterns described in VOID Subsets
<table>
<thead>
<tr>
<th>entity</th>
<th>property</th>
<th>values</th>
</tr>
</thead>
<tbody>
<tr>
<td>descriptor</td>
<td>dct:title</td>
<td>&quot;GVP LOD description document (VOID file)&quot;</td>
</tr>
<tr>
<td>descriptor</td>
<td>dct:created</td>
<td>&quot;2014-03-20&quot;^^xsd:date</td>
</tr>
<tr>
<td>descriptor</td>
<td>dc:format</td>
<td>&quot;meta/void&quot;</td>
</tr>
<tr>
<td>AAT vocab</td>
<td>dct:title</td>
<td>&quot;Art &amp; Architecture Thesaurus (AAT) ®&quot;</td>
</tr>
<tr>
<td>ontology</td>
<td>rdfs:label</td>
<td>&quot;Getty Vocabulary Program ontology&quot;</td>
</tr>
<tr>
<td>ontology</td>
<td>vann:preferredNamespacePrefix</td>
<td>&quot;gvp&quot;</td>
</tr>
<tr>
<td>ontology</td>
<td>vann:preferredNamespaceUri</td>
<td>&quot;<a href="http://vocab.getty.edu/ontology#">http://vocab.getty.edu/ontology#</a>&quot;</td>
</tr>
<tr>
<td>ontology</td>
<td>dc:format</td>
<td>&quot;meta/rdf-schema&quot;</td>
</tr>
<tr>
<td>publisher</td>
<td>foaf:name, rdfs:label</td>
<td>&quot;Getty Research Institute&quot;</td>
</tr>
<tr>
<td>dataset</td>
<td>dcat:contactPoint</td>
<td>[vcard:email <a href="mailto:VocabLOD@getty.edu">mailto:VocabLOD@getty.edu</a>]</td>
</tr>
<tr>
<td>publisher</td>
<td>dct:type</td>
<td><a href="http://purl.org/adms/publishertype/NonProfitOrganisation">http://purl.org/adms/publishertype/NonProfitOrganisation</a></td>
</tr>
<tr>
<td>AAT dataset</td>
<td>dct:title</td>
<td>&quot;AAT Linked Open Data (LOD) Dataset&quot;</td>
</tr>
<tr>
<td>dataset, ontology, exports</td>
<td>dct:created</td>
<td>&quot;2014-02-20&quot;^^xsd:date</td>
</tr>
<tr>
<td>AAT dataset</td>
<td>dcat:keyword</td>
<td>&quot;Thesauri&quot; (similar for TGN)</td>
</tr>
<tr>
<td>AAT dataset</td>
<td>void:exampleResource, adms:sample</td>
<td>aat:300264092, # Objects Facet aat:300264551, # Furnishings and Equipment (Hierarchy Name) aat:300197200, # &lt;containers by function or context&gt; aat:300198841, # Rhyta (similar for TGN, ULAN)</td>
</tr>
<tr>
<td>dataset</td>
<td>dct:accrualPeriodicity</td>
<td>freq:biweekly (frequency of data update and adding new Subjects)</td>
</tr>
<tr>
<td>dataset</td>
<td>dct:source</td>
<td><a href="http://www.getty.edu/research/tools/vocabularies/aat/">http://www.getty.edu/research/tools/vocabularies/aat/</a></td>
</tr>
</tbody>
</table>

4 Only the main AAT languages. Chinese (gvp_lang:zh) covers about 25% of all subjects.
### Getty Vocabularies: LOD

#### Semantic Representation

<table>
<thead>
<tr>
<th>entity</th>
<th>property</th>
<th>values</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataset</td>
<td>dct:type</td>
<td><a href="http://purl.org/adms/assettype/Thesaurus">http://purl.org/adms/assettype/Thesaurus</a></td>
</tr>
<tr>
<td>dataset</td>
<td>adms:interoperabilityLevel</td>
<td><a href="http://purl.org/adms/interoperabilitylevel/Semantic">http://purl.org/adms/interoperabilitylevel/Semantic</a> 5</td>
</tr>
<tr>
<td>dataset</td>
<td>adms:representationTechnique</td>
<td><a href="http://purl.org/adms/representationtechnique/SKOS">http://purl.org/adms/representationtechnique/SKOS</a></td>
</tr>
<tr>
<td>dataset</td>
<td>adms:status</td>
<td><a href="http://purl.org/adms/status/Completed">http://purl.org/adms/status/Completed</a></td>
</tr>
<tr>
<td>AAT vocab, AAT dataset</td>
<td>vann:preferredNamespacePrefix</td>
<td>&quot;aat&quot;</td>
</tr>
<tr>
<td>AAT vocab, AAT dataset</td>
<td>vann:preferredNamespaceUri</td>
<td>&quot;<a href="http://vocab.getty.edu/aat/">http://vocab.getty.edu/aat/</a>&quot;</td>
</tr>
<tr>
<td>AAT dataset</td>
<td>void:uriSpace</td>
<td>&quot;<a href="http://vocab.getty.edu/aat/">http://vocab.getty.edu/aat/</a>&quot;</td>
</tr>
<tr>
<td>dataset, ontology</td>
<td>owl:versionInfo</td>
<td>&quot;3.2&quot;</td>
</tr>
<tr>
<td>dataset</td>
<td>void:feature</td>
<td>fmt:N-Triples, fmt:RDF_XML, fmt:Turtle, fmt:RDF_JSON, fmt:JSON-LD, fmt:SPARQL_Results_XML, fmt:SPARQL_Results_JSON, fmt:SPARQL_Results_CSV, fmt:SPARQL_Results_TSV</td>
</tr>
<tr>
<td>all exports</td>
<td>dc:format</td>
<td>&quot;application/n-triples&quot;, &quot;application/zip&quot;</td>
</tr>
<tr>
<td>explicit exports</td>
<td>dct:title</td>
<td>&quot;Explicit AAT statements&quot;</td>
</tr>
<tr>
<td></td>
<td>dct:description</td>
<td>&quot;NTriples zip, file size is approximate. First load ontologies, then files in indicated order&quot;</td>
</tr>
<tr>
<td></td>
<td>dcat:byteSize</td>
<td>&quot;75000000&quot;^^xsd:decimal</td>
</tr>
<tr>
<td>total exports</td>
<td>dct:title</td>
<td>&quot;Total AAT statements&quot;</td>
</tr>
<tr>
<td></td>
<td>dct:description</td>
<td>&quot;NTriples zip, file size is approximate&quot;</td>
</tr>
<tr>
<td></td>
<td>dcat:byteSize</td>
<td>&quot;80000000&quot;^^xsd:decimal</td>
</tr>
</tbody>
</table>

(*) Notes about dcat:mediaType, dct:format:
- Some VOID examples use MIME type URLs like http://purl.org/NET/mediatypes/application/zip or http://provenanceweb.org/format/mime/application/zip but these URLs don't resolve anymore.
- For some reason "application/n-triples" is not in the IANA Media Types Registry, although it's given in the N-Triples spec. Consequently, this MIME type URL has never resolved.
- We'd like to say void:feature fmt:N-Triples, but this property has domain void:Dataset, while the exports have type dcat:Distribution (object of void:datasetDump).

### 6.5.7 License Info

Unless you are a lawyer, you may not care about licensing info. However, this info is important if you want to ensure you play by the rules, and some software (semantic web crawlers) use it to filter to datasets satisfying certain open data criteria.

---

5 See [European Interoperability Framework (EIF)](https://eif.eu)
6.5.8 Per-Resource Descriptive Info

Alex J Tucker of the BBC suggested in July 2014: "It's important that a machine, when following its nose and retrieving RDF published as Linked Open Data, should be able to tell how the data it fetches is licensed. It's especially important when dealing with the sorts of resources that the Getty vocabularies are applied to". In other words, when a semantic web crawler stumbles upon a GVP resource (e.g. as used in a Cultural Heritage dataset), it should be able to find out the license and dataset of the resource.

We emit such information for all independent resources (see GVP URLs and Prefixes for a list of all prefixes):

<table>
<thead>
<tr>
<th>subject</th>
<th>property</th>
<th>object</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://vocab.getty.edu/dataset">http://vocab.getty.edu/dataset</a></td>
<td>dct:license,</td>
<td><a href="http://opendatacommons.org/licenses/by/1.0/">http://opendatacommons.org/licenses/by/1.0/</a></td>
</tr>
<tr>
<td><a href="http://vocab.getty.edu/dataset/aat">http://vocab.getty.edu/dataset/aat</a></td>
<td>cc:license</td>
<td></td>
</tr>
<tr>
<td><a href="http://vocab.getty.edu/dataset/aat/explicit.zip">http://vocab.getty.edu/dataset/aat/explicit.zip</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="http://vocab.getty.edu/dataset/aat/full.zip">http://vocab.getty.edu/dataset/aat/full.zip</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(same for tgn)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="http://vocab.getty.edu/dataset/aat/tgn">http://vocab.getty.edu/dataset/aat/tgn</a></td>
<td>dct:rights</td>
<td>&quot;Copyright © 2000 The J. Paul Getty Trust. Made available under the ODC Attribution License&quot;</td>
</tr>
<tr>
<td><a href="http://vocab.getty.edu/dataset/aat/tgn">http://vocab.getty.edu/dataset/aat/tgn</a></td>
<td>cc:attributionName</td>
<td>&quot;Contains information from Art &amp; Architecture Thesaurus (AAT)® which is made available under the ODC Attribution License&quot;</td>
</tr>
<tr>
<td></td>
<td>wv:norms</td>
<td><a href="http://www.opendatacommons.org/norms/odc-by-sa/">http://www.opendatacommons.org/norms/odc-by-sa/</a></td>
</tr>
<tr>
<td></td>
<td>wv:declaration</td>
<td>&quot;In circumstances where providing the full attribution statement is not technically feasible, the use of canonical GVP URIs is adequate to satisfy Section 4.3 of the ODC Attribution License&quot;</td>
</tr>
<tr>
<td><a href="http://opendatacommons.org/licenses/by/1.0/">http://opendatacommons.org/licenses/by/1.0/</a></td>
<td>dct:type</td>
<td><a href="http://purl.org/adms/licencetype/Attribution">http://purl.org/adms/licencetype/Attribution</a></td>
</tr>
<tr>
<td></td>
<td>cc:requires</td>
<td>cc:Attribution</td>
</tr>
</tbody>
</table>

6.5.9 VOID Subsets

We declare that the total GVP dataset comprises of the individual vocabulary datasets:

<http://vocab.getty.edu/dataset> void:subset
<http://vocab.getty.edu/dataset/aat>,
<http://vocab.getty.edu/dataset/tgn>,
<http://vocab.getty.edu/dataset/ulan>
As described in Per-Entity Exports, each independent entity (Subject, Source, Contributor) is available in several semantic formats. We express this as void:subsets with uriRegexPattern and the corresponding format:

```xml
  <http://vocab.getty.edu/dataset/aat> void:subset
  <http://vocab.getty.edu/dataset/aat/rdf>
  <http://vocab.getty.edu/dataset/aat/alignment/lcsh>
  dcterms:title "AAT Subjects as RDF/XML";
  void:uriRegexPattern "^http://vocab.getty.edu/aat/\d+.rdf$";
  void:feature fmt:RDF_XML.
```

and similarly for:
- Sources and Contributors

### 6.5.10 VOID Linksets

We also describe the AAT to LCSH Alignment and ULAN to LCNAF Alignment according to VOID section 5 Describing linksets.

```xml
  void:linkPredicate skos:exactMatch.
```

```xml
  void:linkPredicate skos:exactMatch, skos:closeMatch.
```

- AAT-LCSH always uses skos:exactMatch, while ULAN-LCNAF uses the weaker property skos:closeMatch (for imprecise matches like ULAN "Picasso" - LCSH "Picasso in motion pictures") and skos:exactMatch (for precise matches).
- The linksets are provided only as part of the AAT & ULAN datasets and not separately (let us know if you want this changed).
- This is expressed by the void:subset statements. Please note that the void:subset example in section 5.2 Linksets as part of larger datasets uses the wrong direction, see VOID Issue 105
- The alignment is a void:Linkset that connects two void:targets (AAT and LCSH, respectively ULAN and LCNAF) using skos:exactMatch.
- Strictly speaking, http://id.loc.gov/authorities/subjects and http://id.loc.gov/authorities/names are skos:ConceptScheme not void:Datasets. (Note: these pages provides a number of Alternate Formats, but these are "data dumps" or "distributions", not datasets.)
- So if we were purists (we are not), we could describe the target datasets as a blank node, using a trick from VOID section 2.1 Web page links: "As foaf:homepage is an Inverse Functional Property, different descriptions of a dataset provided in different places on the Web can be automatically connected or "smushed" if they use the same homepage URI":

```xml
[a void:Dataset;
  dcterms:title "Library of Congress Subject Headings";
  foaf:homepage <http://id.loc.gov/authorities/subjects>]
```

We also count the void:triples in the linkset, see the end of next section.

### 6.5.11 Dynamic Descriptive Properties

Some of the properties need to be computed dynamically with every update (regeneration) of the dataset. We insert them in named graph http://vocab.getty.edu/well-known/void. We show below only the queries for AAT, similar data is inserted for TGN, ULAN and the overall dataset http://vocab.getty.edu/dataset

- Set dct:modified, dct:issued of dataset, exports, descriptor to the datetime of regeneration (now()). It's safe to assume that some data is updated on every generation; and publication will happen on the same day

```sparql
insert {graph <http://vocab.getty.edu/.well-known/void> {...
```
• Declare all gvp:Facets as void:rootResource of the respective dataset, to let LOD crawlers find all statements in a top-down fashion. This is in lieu of declaring the Top Concepts of AAT

\[
\text{insert } \{ \text{graph } <\text{http://vocab.getty.edu/.well-known/void}> \{ \\
\quad \text{<http://vocab.getty.edu/dataset/aat> void:rootResource ?facet} \\
\quad \} \text{ where } \{ \\
\quad \text{?facet a gvp:Facet; skos:inScheme aat: ;} \\
\quad \} \};
\]

The rest provide VOID statistics (counts) about the datasets. More numbers than you can shake a stick at!

• Count total number of triples

\[
\text{insert } \{ \text{graph } <\text{http://vocab.getty.edu/.well-known/void}> \{ \\
\quad \text{<http://vocab.getty.edu/dataset/aat> void:triples ?count.} \\
\quad \} \text{ where } \{ \\
\quad \text{select (count(*) as ?count) } \{ \text{graph } <\text{http://vocab.getty.edu/dataset/aat}> \{ ?x ?p ?y\}})\\n\quad \} \} \\
\]

• Count number of main entities. We consider only gvp:Subject, not the subsidiary entities

\[
\text{insert } \{ \text{graph } <\text{http://vocab.getty.edu/.well-known/void}> \{ \\
\quad \text{<http://vocab.getty.edu/dataset/aat> void:entities ?count.} \\
\quad \} \text{ where } \{ \\
\quad \text{select (count(*) as ?count) } \{ \text{graph } <\text{http://vocab.getty.edu/dataset/aat}> \{ ?x a ?y\}}. \\
\quad \text{?y rdfs:subClassOf gvp:Subject}})\\n\quad \} \} \\
\]

• Count number of distinct classes

\[
\text{insert } \{ \text{graph } <\text{http://vocab.getty.edu/.well-known/void}> \{ \\
\quad \text{<http://vocab.getty.edu/dataset/aat> void:classes ?count.} \\
\quad \} \text{ where } \{ \\
\quad \text{select (count(*) as ?count) } \{ \text{select distinct ?class } \$GRAPH\{(?x a ?class. \\
\quad \text{filter(!isBlank(?class)))}})\\n\quad \} \} \\
\]

• Count number of distinct properties

\[
\text{insert } \{ \text{graph } <\text{http://vocab.getty.edu/.well-known/void}> \{ \\
\quad \text{<http://vocab.getty.edu/dataset/aat> void:properties ?count.} \\
\quad \} \text{ where } \{ \\
\quad \text{select (count(*) as ?count) } \{ \text{select distinct ?prop } \$GRAPH\{(?x ?prop ?y)}})\\n\quad \} \} \\
\]

• Count number of distinct subjects

\[
\text{insert } \{ \text{graph } <\text{http://vocab.getty.edu/.well-known/void}> \{ \\
\quad \text{<http://vocab.getty.edu/dataset/aat> void:distinctSubjects ?count.} \\
\quad \} \text{ where } \{ \\
\quad \text{select (count(*) as ?count) } \{ \text{select distinct ?subj } \$GRAPH\{(?subj ?p ?y)}})\\n\quad \} \} \\
\]

• Count number of distinct objects

\[
\text{insert } \{ \text{graph } <\text{http://vocab.getty.edu/.well-known/void}> \{ \\
\quad \text{<http://vocab.getty.edu/dataset/aat> void:distinctObjects ?count.} \\
\quad \} \text{ where } \{ \\
\quad \text{select (count(*) as ?count) } \{ \text{select distinct ?obj } \$GRAPH\{(?x ?p ?obj)}})\\n\quad \} \} \\
\]

• Count number of entities per class (class partition). In contrast to void:entities for the whole dataset, here we consider all entities.
The J. Paul Getty Trust  contact us: vocablod@getty.edu

Getty Vocabularies: LOD
Semantic Representation

insert {graph <http://vocab.getty.edu/.well-known/void> {
}} where {
  {select ?class (count(*) as ?count)
   {graph <http://vocab.getty.edu/dataset/aat> {?x a ?class. filter(!isBlank(?class))}} group by ?class};

• Count number of triples per property (property partition)

insert {graph <http://vocab.getty.edu/.well-known/void> {
}} where {
  {select ?prop (count(*) as ?count)
   {graph <http://vocab.getty.edu/dataset/aat> {?x ?prop ?y}} group by ?prop};

• Count number of triples in Linksets (see previous section). The conditions make sure that we count only one direction ?x→?y but not the other direction ?y→?x nor trivial statements ?x→?x and ?y→?y:

# Number of triples in AAT-LCSH Linkset
insert {graph <http://vocab.getty.edu/.well-known/void> {
  <http://vocab.getty.edu/dataset/aat/alignment/lcsh> void:triples ?count
}} where {
  {x skos:exactMatch ?y.
   x skos:inScheme aat:
   filter (strstarts(str(?y),"http://id.loc.gov/"))
}}

# Number of triples in ULAN-LCNAF Linkset
insert {graph <http://vocab.getty.edu/.well-known/void> {
  <http://vocab.getty.edu/dataset/ulan/alignment/lcnaf> void:triples ?count
}} where {
  {x skos:closeMatch ?y.
   x skos:inScheme ulan:
   filter (strstarts(str(?y),"http://id.loc.gov/"))
}}

6.5.12 VOID Deployment
The full descriptive info is available at the following URLs (VOID file in Turtle format):
   • As you see above, this URL is used to tie the descriptor to the other entities
   • This URL is also linked from the GVP LOD home page
   • This URL is implemented as a HTTP 302 redirect to the following URL
In addition, the descriptive info is available
3. In the repository, in its own named graph http://vocab.getty.edu/.well-known/void.
This follows the suggestion "query the endpoint itself in case it indexes its own VoID description" in section 2.1 of SPARQL Web-Querying Infrastructure: Ready for Action? (ISWC 2013) by the creators of the SPARQL Endpoint Status service. It is also queried by RKBExplorer VOID storage (note: this site is down, we've asked the authors for assistance). The following key relations are used to discover the dataset and endpoint:

<http://vocab.getty.edu/.well-known/void>
  foaf:primaryTopic <http://vocab.getty.edu/dataset>.
<http://vocab.getty.edu/dataset>
  void:sparqlEndpoint <http://vocab.getty.edu/sparql>.

You can query the descriptive info using SPARQL, as shown in Counting and Descriptive Info.

6.6 Export Files
We provide export files (data dumps) in several configurations and formats.
6.6.1 Explicit Exports

These are statements in NTriples format, generated from GVP's database using R2RML.

- First load the required External Ontologies (SKOS, SKOS-XL, ISO 25964): links are provided in that section
- Then load the GVP Ontology from http://vocab.getty.edu/ontology.rdf
- Then load the export files:
  - http://vocab.getty.edu/dataset/aat/explicit.zip: 93 Mb zipped, 1561 Mb unzipped (16.8x expansion)
  - http://vocab.getty.edu/dataset/tgn/explicit.zip: 1331 Mb zipped, 27721 Mb unzipped (20.8x expansion)
  - http://vocab.getty.edu/dataset/ulan/explicit.zip: 234 Mb zipped, 4871 Mb unzipped (20.8x expansion)

<table>
<thead>
<tr>
<th>AAT</th>
<th>TGN</th>
<th>ULAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>AATOut_1Subjects.nt</td>
<td>TGNOut_1Subjects.nt</td>
<td>ULANOut_1Subjects.nt</td>
</tr>
<tr>
<td>AATOut_2Terms.nt</td>
<td>TGNOut_2Terms.nt</td>
<td>ULANOut_2Terms.nt</td>
</tr>
<tr>
<td>AATOut_AssociativeRels.nt</td>
<td>TGNOut_AssociativeRels.nt</td>
<td>ULANOut_AssociativeRels.nt</td>
</tr>
<tr>
<td>AATOut_Contribs.nt</td>
<td>TGNOut_Contribs.nt</td>
<td>ULANOut_Contribs.nt</td>
</tr>
<tr>
<td>AATOut_HierarchicalRels.nt</td>
<td>TGNOut_HierarchicalRels.nt</td>
<td>ULANOut_HierarchicalRels.nt</td>
</tr>
<tr>
<td>AATOut_Lang_sameAs.nt</td>
<td>TGNOut_Lang_sameAs.nt</td>
<td>ULANOut_Lang_sameAs.nt</td>
</tr>
<tr>
<td>AATOut_Notations.nt</td>
<td>TGNOut_Notations.nt</td>
<td>ULANOut_Notations.nt</td>
</tr>
<tr>
<td>AATOutObsoleteSubjects.nt</td>
<td>TGNOut_OBSOLETESubjects.nt</td>
<td>ULANOut_OBSOLETESubjects.nt</td>
</tr>
<tr>
<td>AATOut_OrderedCollections.nt</td>
<td>TGNOut_OrderedCollections.nt</td>
<td>ULANOut_OrderedCollections.nt</td>
</tr>
<tr>
<td>AATOutRevisionHistory.nt</td>
<td>TGNOut_RevisionHistory.nt</td>
<td>ULANOut_RevisionHistory.nt</td>
</tr>
<tr>
<td>AATOutRevisionHistorySource.nt</td>
<td>TGNOut_RevisionHistorySource.nt</td>
<td>ULANOut_RevisionHistorySource.nt</td>
</tr>
<tr>
<td>AATOut_ScopeNotes.nt</td>
<td>TGNOut_ScopeNotes.nt</td>
<td>ULANOut_ScopeNotes.nt</td>
</tr>
<tr>
<td>AATOut_SemanticLinks.nt</td>
<td>TGNOut_SemanticLinks.nt</td>
<td>ULANOut_SemanticLinks.nt</td>
</tr>
<tr>
<td>AATOut_Sources.nt</td>
<td>TGNOut_Sources.nt</td>
<td>ULANOut_Sources.nt</td>
</tr>
</tbody>
</table>

- The files inside the zips are named after the different parts of the semantic representation.
- Load them in alphabetical order, for these reasons:
  - Ontotext GraphDB preserves the order of nodes as they are first inserted in the repository, and the first two files are sorted by the required Sort Order
  - Subjects must come before HierarchicalRels because of a [Cut] in Extended Property Constructs.
  - Ensure the required Inference (since these are explicit statements only)
- The above is the actual process we use to load the GVP repository (SPARQL endpoint) with fresh data every 2 weeks.

6.6.2 Per-Entity Exports

The Forest page for each resolvable URL provides downloadable semantic representations for that entity in RDF/XML, Turtle, NTriples, JSON formats. The same formats are available through content negotiation, and through direct URLs including file extension, as described in Semantic Resolution.

For the independent entities (Subjects, Sources, Contributors), the semantic formats include all triples (explicit and inferred) of all owned objects. This information is fetched with complex CONSTRUCT queries and cached for better performance. There is no zip with all these files since they are over 5M: use Total Exports instead

The CONSTRUCT query for the most important kid of entity (Subject) is shown in All Data For Subject. It includes the following triples depicted below:

- All direct triples of the Subject (explicit and inferred) and the other nodes described below
- Local sources (bibo:DocumentPart)
- Terms (skosxl:Label) and scope notes (gvp:ScopeNote) and their local sources
• Change Notes (prov:Activity)
• rdf:Statements describing a relation where the subject plays the role of rdf:subject
• rdf:List nodes of skos:member (i.e. the skos:memberList Structure nodes of an Ordered Array).
• wgs:SpatialThing, schema:Place and schema:GeoCoordinates for TGN concepts
• schema:GeoShape for TGN places that have a bounding box
• schema:Person or schema:Organization for ULAN concepts
• gvp:Biography and bio:Event nodes for ULAN concepts that have them
• rdf:Statements describing the nationality of a ULAN agent

6.6.3 Total Exports
This file includes all statements (explicit and inferred) of all independent entities. It’s a concatenation of the Per-Entity Exports in NTriples format. Because it includes all required Inference, you can load it to any repository (even one without RDFS reasoning):
1. Load the External Ontologies (SKOS, SKOS-XL, ISO 25964): links are provided in that section. The purpose is to get descriptions of properties, associative relations, etc.
2. Load the GVP Ontology from http://vocab.getty.edu/ontology.rdf
3. If your repository supports Subproperty/Inverse/Transitive reasoning:
   • If you want to eliminate the struck-out properties in Reduced SKOS Inference (like we do), execute the SPARQL updates given in Reduced SKOS Inference.
   • If you don’t, these properties will be inferred, which will add 2-3x more statements to your repository.
4. Load the export files:
   • http://vocab.getty.edu/dataset/aat/full.zip: 172 Mb zipped, 2103 Mb unzipped (12.2x expansion)
   • http://vocab.getty.edu/dataset/tgn/full.zip: 2058 Mb zipped, 32880 Mb unzipped (16x expansion)
   • http://vocab.getty.edu/dataset/ulan/full.zip: 381 Mb zipped, 6097 Mb unzipped (16x expansion)

<table>
<thead>
<tr>
<th>AAT</th>
<th>TGN</th>
<th>ULAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>AATOut_Full.nt : subjects</td>
<td>TGNOut_Full.nt : subj, places</td>
<td>ULANOut_Full.nt : subj, agents</td>
</tr>
<tr>
<td>AATOut_Contribs.nt : contributors</td>
<td>TGNOut_Contribs.nt : contributors</td>
<td>ULANOut_Contribs.nt : contributors</td>
</tr>
<tr>
<td>AATOut_Sources.nt : sources</td>
<td>TGNOut_Sources.nt : sources</td>
<td>ULANOut_Sources.nt : sources</td>
</tr>
</tbody>
</table>

• Since the *Out_Full.nt files are very large, they may require some special data loading tool for your repository. Please note that we have not tried out this process yet. If you encounter any problems, please contact us.