# Data Management

## Analysis

**Identify Potential Data Sources -** From spreadsheets to external data repositories, there are a variety of potential data sources that can feed into your VIVO instance. See VIVO Data - what and from where. You'll need to identify which data types and data sources are best aligned with the overall goals for your implementation. See Policy and planning questions for VIVO data. It can take some time to evaluate the data content and quality in order to forecast your ingest strategy. See Data source specifications for implementation and Design of PubmedFetch.

## Design

**Map Data To Ontologies -**The value of the semantic web lies within the ability to define data using widely accepted ontologies. See How to plan data ingest for VIVO. Understanding your data in order to accurately map to the VIVO-ISF ontology is an important step. See VIVO-ISF Ontology. Mapping to an ontology begins as a conceptual design process on a piece of paper or using a diagramming tool such as Vue. Your final data mapping will occur using a tool such as Karma. For more details about using Karma to map your data and produce VIVO compliant semantic web data please visit the Using Karma data integration tool page. Your mapping strategy will not only affect the searchability of your data, but also how easily it can be aggregated and reused by other systems.

**Document Data Cleanup Strategy** - Because linked data has a greater potential to be shared and aggregated, data consistency is crucial. Depending on the quality and variability in your data sources, you may need to plan for data cleanup and/or data "munging" prior to loading into VIVO. See How to manage data cleanup in VIVO. It is also important to standardize and document your data cleanup strategy. Cleanup can be done manually or semi-automated. For a list of suggested cleanup tools see Name disambiguation and entity resolution.

## Implementation

**Prepare Data Loads** - Depending on how your technical roles are defined, this task will be shared between your data staff/domain experts and your developers. This is when ontology mapping goes from conceptual diagrams to actual RDF generation. Preparation of the data loads involves:

- executing the data cleanup strategy
- physically mapping the data to the ontology
- verifying the RDF generated.

See XSLT Ingest Example and Using a different data store

**Document Data Provenance** -Maintenance of the data in your VIVO instance requires thorough documentation of your data flow. Some of this information may be documented internally, while some of it will make sense to load in VIVO as part of the metadata. See Provenance Ontology.

## Launch

**Route Data Cleanup Requests** - From the cleanup strategy identified for the initial ingest, you will need to determine which are ongoing tasks and what the frequency will be going forward. Once your VIVO instance is established, you will want to work with management and to identify the resources and workflow by which data maintenance requests will be addressed. See Data Maintenance. You may have noticed inconsistencies or missing data during your initial ingest that could not be addressed in batch. It is good to be aware of potential cleanup requests and have an established method for correcting the data. See Monitoring for quality.

**Support Data Provisioning** - One of the fun parts of getting your data into VIVO is finding out all of the creative ways it can be reused in other systems. See Finding VIVO Data with the University of Florida Public SPARQL Endpoint. Setting up a SPARQL endpoint is one way to write customized queries for data consumers. See Setting up a VIVO SPARQL Endpoint. SPARQL query results can be run periodically and exported in several common formats such as .csv and .xml. For example queries see Rich export SPARQL queries. Web developers may find the VIVO widgets to be a useful way to consume VIVO data as JSON.

## Maintenance

**Manage Ontology Updates** - Ontologies can be updated to add or remove terms as a domain becomes better defined. Ontology changes can be initialized from within your institution (ex. an identifier specific to your institution) or externally (ex. a deprecated term identified by an international standards organization). The ontology change can be adopted at the level of the the VIVO ontology, or it can be a change done locally within your institution, aka a "local extension". For an example, see Ontology Extensions -- Duke. Local extensions can easily become duplicate ways of defining the same type of information, so it can be helpful to collaborate with the VIVO ontology community when deciding whether or not you need to create a local extension. To the extent that it is relevant to the broader VIVO community, it is important that over time, local ontology extensions get incorporated into the VIVO core ontology.

**Add New Data & Sources** - Data curation is an ongoing task and including new data types or new data sources will most likely be an aspect of maintaining your VIVO instance. You may find that your initial implementation draws more interest from owners of data repositories you hadn't considered. You may want to prioritize new data sources based on the data quality and volume, as well as the number of new ontology mappings required. As an example, see Activating the ORCID integration.