VIVO 1.6 Release Announcement

VIVO 1.6 Release Announcement

VIVO version 1.6 is notable for new features, for significant enhancements to the ontology, and for contributions from developers at institutions beyond the seven partners participating in the 2009-2012 National Institutes of Health VIVO grant. VIVO version 1.6 also marks the first release under the VIVO Incubator Project with DuraSpace (http://duraspace.org).

The VIVO ontology has become VIVO-ISF

The recently completed CTSAconnect project (http://ctsaconnect.org) has refactored, extended, and restructured the VIVO (http://vivoweb.org) and eagle-i (https://www.eagle-i.net) ontologies into an integrated semantic framework (ISF). By combining information about researchers, research resources, and clinical expertise in a single, modular structure, VIVO-ISF provides a more flexible and extensible ontology for both VIVO and eagle-i and for innovative downstream applications consuming this growing pool of richly-structured semantic data, including Plumage (https://github.com/CTSIatUCSF/plumage), VIVOsearch (http://vivosearch.org), and CTSAsearch (http://research.icts.uiowa.edu/polyglot/). While end users of VIVO will experience more continuity than change, the VIVO-ISF ontology aligns more consistently with the Basic Formal Ontology (http://www.ifomis.org/bfo) and better positions VIVO for interoperability with other international ontologies and other semantic web applications. The modular structure of VIVO-ISF also provides adopting institutions more flexibility in determining scope and domain focus for their local context.

In keeping with previous practice, the VIVO 1.6 upgrade script automatically invokes data migration to convert existing VIVO content to the new ontology. VIVO additionally now employs an application configuration ontology allowing more context-dependent control over property appearance and labeling that will be extended in subsequent releases to allow additional customization of VIVO's display and editing behaviors without programming.

Internationalization

VIVO 1.6 offers the option of displaying content, menus, prompts, and ontology labels in alternative and/or multiple languages, a key step in supporting wider VIVO adoption beyond the English-speaking world. By extracting English text from menus, page templates, and the ontology, the language of VIVO's application interface can be fully modified outside of the core code base by copying and translating a small number of files without risk that subsequent releases will overwrite changes. And when multiple language support is enabled by local option, VIVO will respect a user's preferred browser language setting to display the closest matching content as identified by standard RDF language tags.

While VIVO's interface is now language aware, VIVO multiple language support in version 1.6 only extends to public-facing pages and remains read-only except for the ability to add or modify additional primary entity labels in multiple languages. We anticipate expanding support for interactive editing of content in multiple languages in our next release, pending VIVO community feedback and additional user interface design.

Web services

For the first time, VIVO 1.6 exposes data addition, update, and delete actions through an authenticated web service. This service uses SPARQL 1.1 Update, a standard update language for RDF graphs, and allows more seamless interfacing with data ingest tools and allows external applications to read and write arbitrary VIVO data with support for search indexing and associated inferencing. We anticipate that web services will increase the already rapid pace of tool development alongside VIVO providing visualization, editing, reporting, analysis, disambiguation, and repository services. Chris Barnes from the University of Florida is leading a new VIVO Apps & Tools working group (https://wiki.duraspace.org/x/3OEQAg) highlighting existing tools via demos and discussions on biweekly calls.

Performance improvements

Two VIVO community developers have contributed code to support faster page rendering for VIVO when a user is not logged in to edit, allowing VIVO profiles to scale in length and complexity with much less effect on responsiveness. VIVO pages now carry standard HTTP caching headers that web servers and/or more specialized caching libraries can exploit to deliver unchanging content in an instant.

VIVO's search indexing utility has also been extended to support re-indexing a specific subset of data known to have changed, allowing more efficient processing of incremental updates to publications or other content. Responses to linked data requests are faster and more concise, and include a link to the "terms of use" statement provided by the implementing institution.

Look and feel

VIVO keeps the same overall look and feel while sporting a new and more dynamic home page including rotating features highlighting individual research areas, researchers, and departments as well as more prominent statistics on key content elements. An optional map view highlighting the global, national, or regional geographic research focus may also be activated and all features may be customized to local preference.

And many more improvements

In addition to the above major features, VIVO 1.6 includes many new development and debugging features offering implementing sites additional control over deployment, access control, and customization of VIVO pages with additional queries and reports. VIVO's internal SPARQL query endpoint may be configured for authorized access and now supports HTTP content negotiation and JSON-LD.

For additional detail on these and other improvements, please review the full VIVO 1.6 Release Notes (https://wiki.duraspace.org/x/DIRTAg).

New Code Contributors

The VIVO development team gratefully acknowledges code contributions for VIVO 1.6 by Ted Lawless from Brown University, Mark Fallu from Griffith University, and Stephen Williams from the University of Colorado Boulder. We also appreciate extensive bug reports, testing, and feedback from members of the VIVO development, implementation, and ontology working groups. Holly Mistlebauer of Cornell deserves special appreciation for her tireless testing of multiple release candidates, as do Jim Blake, Brian Caruso, Huda Khan, Brian Lowe, and Tim Worrall for addressing 600+ identified issues over the past year of development.