

# 2020-03-24 - VIVO Development IG

## Date

24 Mar 2020

## Call-in Information

Time: 11:00 am, Eastern Time (New York, GMT-04:00)

To join the online meeting:

- Go to: <https://lyrasis.zoom.us/my/vivo1>
- One tap mobile:
  - US: +16699006833,,9358074182# or +19292056099,,9358074182#
- Or Telephone:
  - US: +1 669 900 6833 or +1 929 205 6099 or 877 853 5257
  - Meeting ID: 935 807 4182
- International numbers available: <https://zoom.us/j/aeANHanzED>

## Slack

- <https://vivo-project.slack.com>
  - Self-register at: <http://bit.ly/vivo-slack>

## Attendees

★ Indicating note-taker

1. Don Elsborg
2. Andrew Woods
3. Huda Khan
4. Steven McCauley
5. Benjamin Gross
6. Mike Conlon ★
7. Alexander (Sacha) Jerabek
8. Rachid Belkouch
9. Michel Héon
10. Brian Lowe

## Agenda

1. Sprint planning - i18n
  - a. Sprint dates: Apr 6th - 17th
  - b. Sprint prerequisites
    - i. <https://github.com/UQAM-VIVO>
    - ii. Vivo i18n M2 [http://vivo-i18n.dev.uqam.ca:8080/vivo\\_i18n\\_M2](http://vivo-i18n.dev.uqam.ca:8080/vivo_i18n_M2) (Reference source for testing)
    - iii. ETV Dev Tool <https://github.com/UQAM-VIVO/Vivo-Traduction-Ecosysteme>
  - c. Pre-sprint planning meeting: Mar 25th @10am ET (<https://uqam.zoom.us/j/996861901>)

2. Who is w
  - a.
  - b.
  - c.
  - d. Unable to locate Jira server for this macro. It may be due to Application Link configuration.
3. Using co
  - a.
  - b. Unable to locate Jira server for this macro. It may be due to Application Link configuration.
4. Vitro JMS
5. Incremen
  - a.
  - b. Unable to locate Jira server for this macro. It may be due to Application Link configuration.

- c. Vitro pull
- a. Unable to locate Jira server for this macro. It may be due to Application Link configuration. - looks good

## Tickets

1. Status of In-Review tickets

type	key	summary	assignee	reporter	priority	status	resolution	created	updated	due
Unable to locate Jira server for this macro. It may be due to Application Link configuration.										

## Notes

Draft notes in [Google-Doc](#)

1. Please fill in placeholder page for sprint planning.
  - a. Integration of code into the core. How will this be done?
  - b. Snapshot <https://github.com/uqam-vivo> Running version here [http://vivo-i18n.dev.uqam.ca:8080/vivo\\_i18n\\_M2](http://vivo-i18n.dev.uqam.ca:8080/vivo_i18n_M2)
  - c. Primary goal for the sprint is to get the UQAM code into the core code base. One method is to create a branch of all VIVO codes, then add the UQAM code, then should be ready for pull requests. This should be done before the sprint as sprint preparation.
2. <https://github.com/UQAM-VIVO/Vivo-Traduction-Ecosysteme> a pre-configuration of all the environments needed for VIVO. A complete maven build and run. Installs everything needed. Specially designed for non-full programmers, for people who want to work on translations -- ffl, ontology. For windows. Useful for translators, librarians. Non-developers. UQAM trained 15 people on the environment.
3. 1756 looks good to go. Need to test/review
4. 1755. Turning off the reasoner defeats the admin panel. Now the admin panel can be loaded. Abox reasoner is separate from tbox. There is a visual notification if the reasoner is not running. mostSpecificType is the most likely thing from the reasoner to be missed by people with inferencer turned off. mostSpecificType is commonly used in listViewes to get the right things into lists and interfaces.
5. 1443. Has been a concern for years. Goal is to create a WAR file. Full separation of build and install. One benefit is that a release can include a WAR file and you can just drop it in. vivo home files are packaged into WAR. At start-up it looks to see if you have one, otherwise, boom, it makes one. If you have a vanilla Solr, then perhaps API calls can be made to configure it for Solr. Goal might be to install vanilla tomcat, solr, tdb and then drop a VIVO WAR file in tomcat and everything gets configure, vivo home gets built, all at first time start up.
6. A VIVO archive repo has been created. <https://github.com/vivo-archive> What might be saved? Little? More?
7. 1688 would put the ontology into its own repository. Ontologists could then move forward on major and minor changes.
8. Test framework?
9. What can be our general approach to reducing the dependence of the code on the ontology? The Scholar architecture, for example, isolates ontological change in its mapper, and creates an elastic search index with no references to ontological elements, optimized for presentation purposes.

## Actions

-