

## Mentat - Task #3393

Task # 3374 (Closed): Migrate all core modules from legacy Mentat

Task # 3376 (Closed): Polish mentat-inspector.py module

## Revision of pynspect library

03/25/2017 08:55 PM - Jan Mach

<b>Status:</b>	Closed	<b>Start date:</b>	03/25/2017
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Jan Mach	<b>% Done:</b>	100%
<b>Category:</b>	Development - Core	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	2.0		
<b>Description</b>			
Because mentat-inspector.py is currently being heavily used and expectations are getting bigger, it would be wise to perform revision of the library and finish the implementation, so that everything works as expected and mentat-inspector.py delivers.			
<b>Related issues:</b>			
Related to Mentat - Task # 3392: Revision of JPath library		<b>In Progress</b>	<b>03/25/2017</b>

### Associated revisions

#### Revision ce01f6d7 - 03/29/2017 03:59 PM - Jan Mach

Huge revision of mentat.filtering library.

This commit represents huge revision of mentat.filtering library. Codebase has been checked, unit tests have been improved and documentation has been written to make order in current state of things. Based on this filtering language improvements will be designed. (Redmine issues: #3393, #1017 and #3361)

#### Revision d927033d - 04/04/2017 09:13 AM - Jan Mach

Return None when any operand is None.

(Redmine issue: #3393)

#### Revision f9d35f68 - 04/10/2017 01:31 PM - Jan Mach

Bugfix of None type handling in boolean expressions.

Boolean expressions returned None in case any of their operands was None. This was very counterintuitive and any expression working with variable, that might not exist returned strangely None. (Redmine issue: #3393)

#### Revision aae848f6 - 04/11/2017 09:44 AM - Jan Mach

Bugfix of None type handling in boolean expressions.

Boolean expressions returned None in case any of their operands was None. This was very counterintuitive and any expression working with variable, that might not exist returned strangely None. (Redmine issue: #3393)

#### Revision ac88c6d7 - 05/02/2017 09:15 AM - Jan Mach

Refactoring from mentat.filtering to pynspect.

The mentat.filtering library is being used in another projects separate from Mentat, so it made sense to rename it to something neutral. (Redmine issue: #3393)

#### **Revision efc5e094 - 05/02/2017 10:13 AM - Jan Mach**

Implemented build process for separate pynspect Python packages.

Pynspect library now has separate build process located in deploy/pynspect directory. There is a master makefile. Prior the build current version of library source code is synchronized to local directory, from which the Python packages are then built. The pynspect library still needs to be removed from Mentat package build process. (Redmine issues: #3393 and #3387)

#### **Revision 06b99623 - 05/03/2017 09:58 AM - Jan Mach**

Documentation refactoring for pynspect library.

After refactoring of mentat.filtering library to pynspect the documentation needed update as well. (Redmine issues: #3393 and #3361)

## **History**

---

### **#1 - 03/29/2017 04:55 PM - Jan Mach**

- Assignee changed from Jan Mach to Pavel Kácha

I have made huge revision of filtering library to map the current state. I have tried to document it as well as possible, so we can better plan future improvements. However, there is more pressing work at hand, so we should probably move it to version **Future** for now and come back to it later.

### **#2 - 03/29/2017 05:16 PM - Jan Mach**

- Related to Task #3392: Revision of JPath library added

### **#3 - 03/30/2017 10:40 AM - Pavel Kácha**

- Assignee changed from Pavel Kácha to Jan Mach

- Priority changed from High to Normal

- Target version changed from 2.0 to Future

Yep, will need revisiting. Some of the operators are not consistent, were created as ad-hoc solutions, some applies for return types (especially values vs lists).

Food for thought:

- type coercion (if it's not list, make it one) vs explicit return types?
- less operators doing "the right thing" or more specific ones (scalar, list, set)
- syntax freedom (how clever to be in recognizing IPs, times, strings)?
- working with specific types
  - list if IP addresses is list of lists? list of ranges? what about operations?
  - how about time operations
- what to do with operation on nonexistent keys? Need to riddle queries with explicit "exists" calls, or somehow try to do the right thing? Is 3 > nonexistent? nonexistent > 3?

Also:

- review existing operator implementations, especially that they use current idioms (one of the suspects is "exists" not calling proper jsonpath)

Let's get inspiration from Python, but we have a special case, so let's keep it simple and translatable even to database languages. That might even mean that sometimes we will have to say that some result is undefined (db would do something else than direct code, or db does not support).

**#4 - 05/02/2017 09:20 AM - Jan Mach**

- *Subject changed from Revision of mentat.filtering library to Revision of pynspect library*

The library was renamed from **mentat.filtering** to **pynspect**, because it is being used outside of Mentat project it made sense for it to have a neutral name.

**#5 - 10/03/2017 07:12 PM - Jan Mach**

- *Status changed from In Progress to Closed*
- *% Done changed from 0 to 100*

The pynspect library is now distributed as separate package.

**#6 - 07/27/2018 11:24 AM - Jan Mach**

- *Target version changed from Future to 2.0*