

Ruby master - Feature #7788

YAML Tag Schema Support

02/06/2013 05:05 AM - trans (Thomas Sawyer)

Status:	Open
Priority:	Normal
Assignee:	tenderlovmaking (Aaron Patterson)
Target version:	
Description	
<p>=begin</p> <p>I have endeavoured to add proper Schema support to Psych (see (()) on Schemas). The primary reasons for supporting schemas are two fold: security and global tag conflict. The first is well known b/c of recent events. The second is less realized, but consider it is same problem as using global variables. Different apps have different tags; two identical local tags may have different meanings and thus cause conflict.</p> <p>The API works like this:</p> <pre>class Foo end foo_schema = YAML::Schema.new do s s.tag '!foo', Foo end YAML.load('foo.yml', :schema=>foo_schema)</pre> <p>This code would allow only failsafe and json schema tags (core defaults), plus the specifically defined !foo tag. Also, %TAG prefix is supported:</p> <pre>foo_schema = YAML::Schema.new(:prefix=>{'!'=>'tag:foo.org/'}) do s s.tag '!foo', Foo end</pre> <p>This will add tag 'tag:foo.org/fooinstead of local!foo` tag.</p> <p>To properly support schema, object's must store the tag with which they were loaded in order to ensure correct round tripping. For this there is tag_uri attribute. (Note: I am not sure if it best to store as instance variable, which it currently is, or to store in global table. Need feedback.)</p> <p>In the process of adding schema support I was able to clean up and generalize loading code. For immutable types and class factories, adding ((ClassName.new_with(coder))) can be used to instantiate class.</p> <p>Implementation is close to complete, I believe this is all that remains:</p> <ol style="list-style-type: none">1. ScalarScanner needs to respect schema (basically if failsafe and/or json schemas are not used).2. Dumping needs to take :schema option to limit it to schema tags.3. Dumping needs to look to tag_uri for tag by default.4. There is one bug I have yet to figure out (test_spec_builtin_map).5. I have questions about Coder, b/c it seems more complex than it needs to be. <p>I am also considering refactoring Schemas as modules that can be included into other schema. Currently they are classes/objects that can be subclassed or merged via +, e.g.</p> <pre>LEGACY_SCHEMA = CORE_SCHEMA + RUBY_SCHEMA + OBJECT_SCHEMA + SYCK_SCHEMA</pre> <p>Of course, as with any new code, there's sure to be corner cases to work out. Having other pound on it for a while would be very helpful. Oh, and I should also mention I am documenting as much of the code as can.</p> <p>Feel free to ask me any questions for more details about the code. You can find the branch here: https://github.com/trans/psych/tree/isotag</p> <p>=end</p>	

History

#1 - 02/22/2013 09:09 AM - ko1 (Koichi Sasada)

- Assignee set to *tenderlovmaking* (Aaron Patterson)

#2 - 12/25/2017 06:15 PM - naruse (Yui NARUSE)

- Target version deleted (2.6)