Ruby master - Bug #15764

Whitespace and control characters should not be permitted in tokens

04/11/2019 08:59 PM - BatmanAoD (Kyle Strand)

Status:	Open		
Priority:	Normal		
Assignee:	matz (Yukihiro Matsumoto)		
Target version:			
ruby -v:		**Backport:**	2.4: UNKNOWN, 2.5: UNKNOWN, 2.6: UNKNOWN

Description

As of Ruby 2.5.1p57, it appears that all valid Unicode code-points above 128 are permitted in tokens. This includes whitespace and control characters.

This was demonstrated here: https://gist.github.com/qrohlf/7045823

I have attached the raw download from the above gist.

The issue has been discussed on StackOverflow: https://stackoverflow.com/q/34455427/1858225

I would say this is arguably a bug, but I am marking this ticket as a "feature" since the current behavior could be considered by-design.

History

#1 - 04/13/2019 06:18 AM - shevegen (Robert A. Heiler)

I am not sure this is a bug, though. If it is a valid Unicode token then it should work, so the behaviour would seem correct to me. Actually sawa made a similar comment on SO - he is very active on the ruby bug tracker here too. :-)

As to whether it is a feature or not, hmm - I guess it depends on the definition of whitespace as far as Unicode is concerned (and then on ruby whether it agrees with this definition). Perhaps Durst could comment as to whether he knows of any unicode character(s) that should be treated as one that should not be allowed to use for a variable (e. g. when ruby decided to treat it as whitespace).

I am, however had, not sure if this is anything that is really highly important - the SO link shows someone just having fun, with the cryptically written "Hello world". Are many people using non-ASCII Unicode in .rb files these days for real work? I myself am way too oldschool to want to transition away from ASCII/ISO* (in europe at the least).

#2 - 04/13/2019 09:51 AM - duerst (Martin Dürst)

- Backport set to 2.4: UNKNOWN, 2.5: UNKNOWN, 2.6: UNKNOWN

- Assignee set to matz (Yukihiro Matsumoto)

- Tracker changed from Feature to Bug

I also think this is a bug. I have changed the category accordingly.

I think we should restrict the characters usable in identifiers to some reasonable ranges. I agree that we mainly want to focus on ASCII programs, but we should do at least a sanity check for the rest of Unicode, and that's clearly not happening now.

As a base for this, it's best to look at Unicode Standard Annex #31, Unicode Identifier And Pattern Syntax (http://www.unicode.org/reports/tr31/). A regular expression for the identifier syntax defined in UAX #31 is easily available in Ruby: /\p{id_start}\p{id_continue}*/. The character ranges covered by these properties can be checked in enc/unicode/12.1.0/name2ctype.h, from lines 15267 and 15881 (the file is too large for the Web interface to svn).

The only additions we seem to need are '_' in initial position, sigils for the different kinds of identifiers, and final '!', '?', and '=' for method names.

I suspect that it may take nobu (Nobuyoshi Nakada) just a few hours to actually implement this, and that the backwards-compatibility issues (existing Ruby programs stopping to work) are extremely minimal and limited to examples that show the problem.

I have added this to the list of issues to be discussed at next week's developers' meeting, but I will not be at the meeting itself. If needed, I can join the discussion at the first day of RubyKaigi itself. I have assigned this issue to Matz because I'd like him to give it a sanity check.

#3 - 04/16/2019 10:05 AM - duerst (Martin Dürst)

There may be a question as to what to do with encodings other than UTF-8. I see three possibilities:

1) Ignore (i.e. leave as is), because who still uses them?

2) Use a rule based on character properties such as letter and digit that are supported in these encodings

3) Convert to Unicode and check there

I would be okay with any one of these.

#4 - 04/21/2019 02:40 AM - duerst (Martin Dürst)

Some points from a discussion at the Fukuoka post-RubyKaigi event:

- It might make sense to allow emoji as variable names (just for fun)
- Python also limits non-ASCII identifiers. See https://docs.python.org/3/reference/lexical_analysis.html#identifiers. What they do is based on UAX #31, and applies NFKC during parsing.

#5 - 04/22/2019 07:48 AM - naruse (Yui NARUSE)

Whether an issue is "Bug" or "Feature" practically depends whether the fix should be backported or not.

Anyway as far as I understand, Ruby has following categories of characters.

- First character for constants (upper case)
- First character for local variables (lower case and '_')
- trailing character for tokens (upper case, lowercase, '_', and numerics)
- white spaces
- sigils (ASCII punct except '_')
- invalid char (non nul non space control characters)

So we need to map between Unicode category and above category.

In addition we may also consider the canonicalize for each sigils like !, ?, and so on.

Files

helloworld.rb	543 Bytes	04/11/2019	BatmanAoD (Kyle Strand)