The methods are named to be consistent with MatchData.
My C-fu isn't very strong, so a review of the patch would be appreciated.
I attached the revised code and a diff file. If other formats are wished, please tell.
If I need to submit something else with that patch, please tell.

Regards
Stefan
strscan_aref(VALUE self, VALUE idx)

/*
 * call-seq: size
 *
 * Return the amount of subgroups in the most recent match.
 * The full match counts as a subgroup.
 *
 * s = StringScanner.new("Fri Dec 12 1975 14:39")
 * s.scan(/(\w+) (\w+) (\d+) /)    # -> "Fri Dec 12 "
 * s.size                          # -> 4
 */

static VALUE
strscan_size(VALUE self)
{
    struct strscanner *p;

    GET_SCANNER(self, p);
    if (!MATCHED_P(p)) return Qnil;
    return INT2FIX(p->regs.num_regs);
}

/*
 * call-seq: captures
 *
 * Returns the subgroups in the most recent match (not including the full match).
 * If nothing was priorly matched, it returns nil.
 *
 * s = StringScanner.new("Fri Dec 12 1975 14:39")
 * s.scan(/(\w+) (\w+) (\d+) /)    # -> "Fri Dec 12 "
 * s.captures                      # -> ["Fri", "Dec", "12"]
 * s.captures                      # -> nil
 */

static VALUE
strscan_captures(VALUE self)
{
    struct strscanner *p;
    int   i, num_regs;
    VALUE new_ary;

    GET_SCANNER(self, p);
    if (!MATCHED_P(p)) return Qnil;

    num_regs = p->regs.num_regs;
    new_ary = rb_ary_new2(num_regs);
    for (i = 1; i < num_regs; i++) {
        VALUE str = extract_range(p, p->prev + p->regs.beg[i],
            p->prev + p->regs.end[i]);
        rb_ary_push(new_ary, str);
    }
    return new_ary;
}

/*
 * call-seq:
 *      scanner.values_at( i1, i2, ... iN ) -> an_array
 *
 * Returns the subgroups in the most recent match at the given indices.
 * If nothing was priorly matched, it returns nil.
 *
 * s = StringScanner.new("Fri Dec 12 1975 14:39")
 * s.scan(/(\w+) (\w+) (\d+) /)    # -> "Fri Dec 12 "
 * s.values_at 0, -1, 5, 2         # -> ["Fri Dec 12 ", "12", nil, "Dec"]
 */

static VALUE
strscan_values_at(VALUE self, VALUE idx)
{
+strscan_values_at(int argc, VALUE *argv, VALUE self) +{ +struct strscanner *p; +long i; +VALUE new_ary; + +GET_SCANNER(self, p); +if (! MATCHED_P(p)) return Qnil; + +new_ary = rb_ary_new2(argc); +for (i = 0; i<argc; i++) { + rb_ary_push(new_ary, strscan_aref(self, argv[i])); +} + +return new_ary; +} + */ + */ * Return the <i><b>pre</b>-match</i> (in the regular expression sense) of the last scan. * @@ -1312,4 +1398,7 @@ Init_strscan(void) + rb_define_method(StringScanner, "size", strscan_size, 0); + rb_define_method(StringScanner, "captures", strscan_captures, 0); + rb_define_method(StringScanner, "values_at", strscan_values_at, -1); + rb_define_method(StringScanner, "rest", strscan_rest, 0); +} + }
Seems fine to me.

#9 - 11/24/2012 01:11 PM - mame (Yusuke Endoh)
- Target version changed from 2.0.0 to 2.6

#10 - 11/29/2017 07:57 AM - nobu (Nobuyoshi Nakada)
- Status changed from Assigned to Closed

Applied in changeset trunk/r60929.

---

strscan.c: add MatchData-like methods

- ext/strscan/strscan.c: added size, captures and values_at to StringScanner, shorthands of accessing the matched data. based on the patch by apeiros (Stefan Rusterholz) at [ruby-core:20412]. [Feature #836]

Files

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>strscan.c</td>
<td>38.2 KB</td>
<td>12/08/2008</td>
<td>apeiros (Stefan Rusterholz)</td>
</tr>
<tr>
<td>strscan.diff</td>
<td>3.12 KB</td>
<td>12/08/2008</td>
<td>apeiros (Stefan Rusterholz)</td>
</tr>
</tbody>
</table>