

```
1: // $Id: template_string.cpp,v 1.4 2016-08-02 17:51:49-07 - - $
2:
3: #include <iostream>
4: using namespace std;
5:
6: template <typename char_type>
7: struct template_string {
8:
9:     struct long_str {
10:         size_t capacity_;
11:         size_t size_;
12:         char_type* data_;
13:     };
14:
15:     static constexpr size_t char_count = (sizeof (long_str) - 1)
16:                                           / sizeof (char_type);
17:     static constexpr size_t min_cap = char_count > 2 ? char_count : 2;
18:
19:     struct short_str {
20:         union {
21:             unsigned char size_;
22:             char_type padding_;
23:         };
24:         char_type data_[min_cap];
25:     };
26:
27:     union str_rep {
28:         long_str long_;
29:         short_str short_;
30:     } rep_;
31:
32:     bool is_long_str() const {
33:         return rep_.short_.size_ & 1;
34:     }
35:
36: };
37:
```

```
38:
39: int offset (const void* base, const void* field) {
40:     return static_cast<const char*> (field)
41:         - static_cast<const char*> (base);
42: }
43:
44: #define SHOW(X) cout << #X << " = " << X << endl;
45: template <typename char_type_t>
46: void print_info (const string& char_type) {
47:     template_string<char_type_t> v;
48:     SHOW (char_type);
49:     SHOW (sizeof (char_type_t));
50:     SHOW (sizeof v);
51:     SHOW (sizeof v.rep_);
52:     SHOW (sizeof v.rep_.long_);
53:     SHOW (sizeof v.rep_.long_.capacity_);
54:     SHOW (sizeof v.rep_.long_.size_);
55:     SHOW (sizeof v.rep_.long_.data_);
56:     SHOW (sizeof v.rep_.short_);
57:     SHOW (sizeof v.rep_.short_.size_);
58:     SHOW (sizeof v.rep_.short_.padding_);
59:     SHOW (sizeof v.rep_.short_.data_);
60:     SHOW (v.char_count);
61:     SHOW (v.min_cap);
62:     SHOW (offset (&v, &v));
63:     SHOW (offset (&v, &v.rep_));
64:     SHOW (offset (&v, &v.rep_.long_));
65:     SHOW (offset (&v, &v.rep_.long_.capacity_));
66:     SHOW (offset (&v, &v.rep_.long_.size_));
67:     SHOW (offset (&v, &v.rep_.long_.data_));
68:     SHOW (offset (&v, &v.rep_.short_));
69:     SHOW (offset (&v, &v.rep_.short_.size_));
70:     SHOW (offset (&v, &v.rep_.short_.padding_));
71:     SHOW (offset (&v, &v.rep_.short_.data_));
72: }
73:
74: #define PRINT(C) print_info<C>(#C);
75: int main() {
76:     PRINT(char);
77:     cout << "\f" << endl;
78:     PRINT(wchar_t);
79:     cout << "\f" << endl;
80:     PRINT(char16_t);
81:     cout << "\f" << endl;
82:     PRINT(char32_t);
83: }
84:
85: /*
86: //TEST// ./template_string >template_string.out
87: //TEST// mkpspdf template_string.ps template_string.cpp* \
88: //TEST//      template_string.out
89: */
90:
```

[illegible]

```
1: char_type = char
2: sizeof (char_type_t) = 1
3: sizeof v = 24
4: sizeof v.rep_ = 24
5: sizeof v.rep_.long_ = 24
6: sizeof v.rep_.long_.capacity_ = 8
7: sizeof v.rep_.long_.size_ = 8
8: sizeof v.rep_.long_.data_ = 8
9: sizeof v.rep_.short_ = 24
10: sizeof v.rep_.short_.size_ = 1
11: sizeof v.rep_.short_.padding_ = 1
12: sizeof v.rep_.short_.data_ = 23
13: v.char_count = 23
14: v.min_cap = 23
15: offset (&v, &v) = 0
16: offset (&v, &v.rep_) = 0
17: offset (&v, &v.rep_.long_) = 0
18: offset (&v, &v.rep_.long_.capacity_) = 0
19: offset (&v, &v.rep_.long_.size_) = 8
20: offset (&v, &v.rep_.long_.data_) = 16
21: offset (&v, &v.rep_.short_) = 0
22: offset (&v, &v.rep_.short_.size_) = 0
23: offset (&v, &v.rep_.short_.padding_) = 0
24: offset (&v, &v.rep_.short_.data_) = 1
```

```
25:
26: char_type = wchar_t
27: sizeof (char_type_t) = 4
28: sizeof v = 24
29: sizeof v.rep_ = 24
30: sizeof v.rep_.long_ = 24
31: sizeof v.rep_.long_.capacity_ = 8
32: sizeof v.rep_.long_.size_ = 8
33: sizeof v.rep_.long_.data_ = 8
34: sizeof v.rep_.short_ = 24
35: sizeof v.rep_.short_.size_ = 1
36: sizeof v.rep_.short_.padding_ = 4
37: sizeof v.rep_.short_.data_ = 20
38: v.char_count = 5
39: v.min_cap = 5
40: offset (&v, &v) = 0
41: offset (&v, &v.rep_) = 0
42: offset (&v, &v.rep_.long_) = 0
43: offset (&v, &v.rep_.long_.capacity_) = 0
44: offset (&v, &v.rep_.long_.size_) = 8
45: offset (&v, &v.rep_.long_.data_) = 16
46: offset (&v, &v.rep_.short_) = 0
47: offset (&v, &v.rep_.short_.size_) = 0
48: offset (&v, &v.rep_.short_.padding_) = 0
49: offset (&v, &v.rep_.short_.data_) = 4
```

```
50:
51: char_type = char16_t
52: sizeof (char_type_t) = 2
53: sizeof v = 24
54: sizeof v.rep_ = 24
55: sizeof v.rep_.long_ = 24
56: sizeof v.rep_.long_.capacity_ = 8
57: sizeof v.rep_.long_.size_ = 8
58: sizeof v.rep_.long_.data_ = 8
59: sizeof v.rep_.short_ = 24
60: sizeof v.rep_.short_.size_ = 1
61: sizeof v.rep_.short_.padding_ = 2
62: sizeof v.rep_.short_.data_ = 22
63: v.char_count = 11
64: v.min_cap = 11
65: offset (&v, &v) = 0
66: offset (&v, &v.rep_) = 0
67: offset (&v, &v.rep_.long_) = 0
68: offset (&v, &v.rep_.long_.capacity_) = 0
69: offset (&v, &v.rep_.long_.size_) = 8
70: offset (&v, &v.rep_.long_.data_) = 16
71: offset (&v, &v.rep_.short_) = 0
72: offset (&v, &v.rep_.short_.size_) = 0
73: offset (&v, &v.rep_.short_.padding_) = 0
74: offset (&v, &v.rep_.short_.data_) = 2
```

```
75:
76: char_type = char32_t
77: sizeof (char_type_t) = 4
78: sizeof v = 24
79: sizeof v.rep_ = 24
80: sizeof v.rep_.long_ = 24
81: sizeof v.rep_.long_.capacity_ = 8
82: sizeof v.rep_.long_.size_ = 8
83: sizeof v.rep_.long_.data_ = 8
84: sizeof v.rep_.short_ = 24
85: sizeof v.rep_.short_.size_ = 1
86: sizeof v.rep_.short_.padding_ = 4
87: sizeof v.rep_.short_.data_ = 20
88: v.char_count = 5
89: v.min_cap = 5
90: offset (&v, &v) = 0
91: offset (&v, &v.rep_) = 0
92: offset (&v, &v.rep_.long_) = 0
93: offset (&v, &v.rep_.long_.capacity_) = 0
94: offset (&v, &v.rep_.long_.size_) = 8
95: offset (&v, &v.rep_.long_.data_) = 16
96: offset (&v, &v.rep_.short_) = 0
97: offset (&v, &v.rep_.short_.size_) = 0
98: offset (&v, &v.rep_.short_.padding_) = 0
99: offset (&v, &v.rep_.short_.data_) = 4
```