# Freeciv - Bug #854307

## Can't use bare string as printf format string with Q_() and friends

2019-12-30 11:51 AM - Jacob Nevins

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
<th>Start date:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
<td>Due date:</td>
</tr>
<tr>
<td>Assignee:</td>
<td>Jacob Nevins</td>
<td>% Done:</td>
</tr>
<tr>
<td>Category:</td>
<td></td>
<td>0%</td>
</tr>
<tr>
<td>Sprint/Milestone:</td>
<td>2.6.2</td>
<td>Estimated time:</td>
</tr>
</tbody>
</table>

**Description**

On my system (Debian buster, GCC 8.3.0), I'm allowed to write something like

```c
cat_snprintf(buf, bufsz, _("Fixed string"));
```

but if I change it to

```c
cat_snprintf(buf, bufsz, Q_("?qual:Fixed string"));
```

I get this warning:

```plaintext
error: format not a string literal and no format arguments [-Werror=format-security]
```

I think this is because `_()` (i.e. the system gettext()) benefits from GCC attribute `__attribute__ ((__format_arg__))`, but our homegrown `Q_()` and friends do not. So I think we should add that attribute to our `skip_intl_qualifier_prefix()`.

**About GCC version support:** gettext's rune is defined in a `cdefs.h` on my system as follows:

```c
/* At some point during the gcc 2.8 development the `format_arg' attribute for functions was introduced. We don't want to use it unconditionally (although this would be possible) since it generates warnings. If several `format_arg' attributes are given for the same function, in gcc-3.0 and older, all but the last one are ignored. In newer gccs, all designated arguments are considered. */
#if __GNUC_PREREQ (2,8)
#define __attribute_format_arg__(x) __attribute__ ((__format_arg__ (x)))
#else
#define __attribute_format_arg__(x) /* Ignore */
#endif
```

Those versions are pretty old (GCC 3.0 was 2001). I haven't found documentation for a minimum version of GCC we support, but it looks like we use the `__attribute__` attribute without checking GCC version? I don't know how old that attribute is.

For now I'm not going to try to conditionalise this on GCC version. (And I'm going to assume that any toolchain which defines `__GNUC__` can cope with them, as we do for other attributes -- I haven't tried e.g. clang.)

**History**

#1 - 2019-12-30 01:08 PM - Jacob Nevins
- File `m-30-26-q-attr-format-string.patch` added
- Status changed from In Progress to Resolved

#2 - 2019-12-30 03:39 PM - Marko Lindqvist
I think we currently have some `printf_type_function("%s", Q_("?qual:Fixed string"))` kind of calls in our code to work around this.

#3 - 2019-12-30 05:13 PM - Marko Lindqvist
Jacob Nevins wrote:

(And I'm going to assume that any toolchain which defines `__GNUC__` can cope with them, as we do for other attributes -- I haven't tried e.g.)
Defining __GNUC__ is their own claim of compatibility. Anyway, your patch compiles fine with clang-5 and clang-9 (clang version extremes on my main development machine, didn't bother to run any virtual machine to test with more ancient versions) Tested on S2_6.

#4 - 2020-01-01 03:17 AM - Jacob Nevins

- Status changed from Resolved to Closed

Files

<table>
<thead>
<tr>
<th>File Name</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>m-30-26-q-attr-format-string.patch</td>
<td>959 Bytes</td>
<td>2019-12-30</td>
<td>Jacob Nevins</td>
</tr>
</tbody>
</table>