For those locale constructors that take a pointer, it is not
documented what happens if the pointer is 0. Also, for those
constructors that take another locale and copy facets from it,
it is not specified what happens if it is the transparent
locale.

[this is an Editorial Proposal.]

They should throw something.

The LWG discussed the matter at Santa Cruz, and decided on all cases,
as follows. Class locale has three constructors affected by this
resolution:

```cpp
explicit locale(const char* std_name);
locale(const locale& other, const char* std_name, category)

template <class Facet>
locale(const locale& other, Facet* f);
```

For the first two, the LWG decided it should throw `runtime_error`
if the const char* argument is null. For the third, it decided
that if the Facet* argument is null, the constructed locale is
identical to the const locale& argument.

** Proposed Resolution:

In [lib.locale.cons] (22.1.1.2, p. 22-7), in the definition of

```cpp
explicit locale(const char* std_name);
```

replace the last sentence of the Effects: paragraph with: "Throws
runtime_error if the argument is not valid, or is null."

In the definition of

```cpp
locale(const locale& other, const char* std_name, category);
```

add the quoted sentence above to the Effects: clause paragraph.

In the definition of the template constructor

```cpp
template <class Facet> locale(const locale& other, Facet* f);
```

add to the Effects: paragraph: "If *f* is null, the resulting locale
is a copy of argument *other*."