
Subject: Re: Question about SubRange.

Posted by [mirek](#) on Mon, 08 Jan 2018 10:52:02 GMT

[View Forum Message](#) <> [Reply to Message](#)

Novo wrote on Thu, 04 January 2018 21:58Function SubRange has two overloads below.

```
template <class C>
auto SubRange(C& c, int pos, int count) -> decltype(SubRange(c.begin() + pos, count))
{
    return SubRange(c.begin() + pos, count);
}
```

```
template <class C>
auto SubRange(C&& c, int pos, int count) -> decltype(SubRange(c.begin() + pos, count))
{
    return SubRange(c.begin() + pos, count);
}
```

The second one doesn't actually move anything, so, it is just not needed, IMHO.

The first one can always be used instead. And it should look like below. IMHO.

```
template <class C>
auto SubRange(const C& c, int pos, int count) -> decltype(SubRange(c.begin() + pos, count))
{
    return SubRange(c.begin() + pos, count);
}
```

Method begin() is always const I believe ...

Not true (not in U++ nor STL). That is basically a reason for those 2 overloads. E.g. try Sort with just const variant...

Quote:

And you can drop this decltype in C++14 ...

Not if we want to maintain C++11 compatibility (current status is that we do and we will as long as it is cheap. I think decltype here is cheap).
