Subject: Re: 2024rc1
Posted by Lance on Mon, 21 Oct 2024 16:12:11 GMT
View Forum Message <> Reply to Message

mirek wrote on Mon, 21 October 2024 08:45Lance wrote on Mon, 21 October 2024 13:01

A compiler that encounters a non-constexpr if-statement with a constexpr conditional may issue a warning advising you to use if constexpr instead. This will ensure that compile-time evaluation will occur (even if optimizations are disabled).

Well, thinking about it, I guess actually the real benefit for me would be something else: compiler issues an error if the expression you marked constexpr is not...

Agreed. Like "override", make a programmer's intention more explicit, and do, more important than but similar, compiler check when it doesn't going as claimed by the programmer.

Reminds me of a related case, where constexpr seems to be helpful or possibly necessary.

```
union Flags{
 int32 dummy;
 struct{
  byte borderLeft  :3;
  byte borderRight :3;
  byte borderTop   :3;
  byte borderBottom:3;
  byte halign      :2;
  byte valign      :2; //16th bit

  bool faceNotNull     :1;
  bool boldNotNull     :1;
  bool heightNotNull   :1;
  bool widthNotNull    :1;
  bool underlineNotNull:1;
  bool italicNotNull   :1;
  bool strikeoutNotNull:1; //23rd bit
 };

 constexpr Flags() : dummy(0){ static_assert(sizeof(*this)==sizeof(dummy)); }


 static constexpr int32 FontMask()
 {
  Flags f;
  f.faceNotNull = true;
  f.boldNotNull = true;
  f.heightNotNull = true;
```

```
  f.widthNotNull = true;
  f.underlineNotNull = true;
  f.italicNotNull = true;
  f.strikeoutNotNull = true;
  return f.dummy;
 }
};
```

It's a somewhat contrived example. I am not sure ,for int32 FontMask(), if I don't it constexpr, will the code be compiled same as if I do. It's totally possible they do with todays smart and agressive as crazy compiler optimization.
