Subject: Re: Drop list is not shown
Posted by Mindtraveller on Tue, 18 Nov 2014 23:21:15 GMT
View Forum Message <> Reply to Message

I've finally found the source of problem.

Second window turned out to be false path. It had nothing to do with dropping. After some testing it became clear that drop list failure doesn't depend on window, but it depend on whether specific control is shown in window. Right after control is hidden, all drop lists are fired one after another. This control is just slightly customized version on Edit control with additional buttons. As later testing revealed, the problem was not with the control itself, but with custom Convert used in it. However I'll post code of this control just to show what I'm talking about:

```
class EditIntKeyless : public EditInt
{
public:
 EditIntKeyless()
  :convertChecker(this)
 {
  mbutton.AddButton().SetImage(IMG::b_minus).Width(23) <<= callback(this,
&EditIntKeyless::OnDec);
  mbutton.AddButton().SetImage(IMG::b_plus) .Width(23) <<= callback(this,
&EditIntKeyless::OnInc);
  AddFrame(mbutton);
  SetData(1);
  this->WantFocus(true);
  SetConvert(static_cast<Convert &>(convertChecker));
 }
 void SetupChecker(ChannelChecker *c, int a)
 {
  convertChecker.SetupChecker(c,a);
  convertChecker.Check(GetData());
 }
 virtual void SetData(const Value& data)
 {
  convertChecker.Check(data);
  EditInt::SetData(data);
 }

private:
 void OnInc()   {SetData( 1 + (int) GetData()); WhenAction();}
 void OnDec()   {SetData(-1 + (int) GetData()); WhenAction();}
 MultiButtonFrame mbutton;
 ConvertChecker convertChecker;
};
```

As you may see, this control supports some kind of "channel checker" (for us, it is not that important what is that) which gets control's pointer.

The problem arises when custom Convert considers control value invalid and tries to show it to user by calling Edit::Error. If you're interested, let's take a closer look at its source, it is short too:

```
class ConvertChecker : public ConvertInt
{
public:
 ConvertChecker(EditField *_edit)
  :checker(NULL)
  ,addr(-1)
  ,channel(-1)
  ,edit(_edit)
 {}
 void SetupChecker(ChannelChecker *c, int a) {checker = c; addr = a;}

public:
 virtual Value Scan(const Value& text) const
 {
  Value chV = ConvertInt::Scan(text);
  int   ch = chV;
  Check(ch);
  return chV;
 }

 void Check(int64 c) const
 {
  if (channel >= 0)
   UnregisterChannel(channel);
  channel = static_cast<int>(c);

  bool isFree = CheckChannel(channel);
  if (channel >= 0)
   RegisterChannel(channel);
  //edit->Error(!isFree); // <-- this is the source of the problem!
 }

private:
 bool CheckChannel(int ch) const
 {
  if (!checker || addr < 0 || ch < 0)
   return true;
  return checker->IsChannelFree(addr,ch);
 }
 void RegisterChannel(int ch) const
 {
  if (!checker || addr < 0)
   return;
  checker->RegisterChannel(addr,ch);
 }
 void UnregisterChannel(int ch) const
 {
  if (!checker || addr < 0)
   return;
  checker->UnregisterChannel(addr,ch);
 }

 ChannelChecker *checker;
 int            addr;
 mutable int    channel;
 EditField      *edit;
};
```

Here we have edit->Error() call which makes all the problems. If commented, all the drop lists work flawlessly. I haven't looked deeper into U++ code, but it looks like erroneous state of control stalls any dropping animations in GUI.
I don't know if this is a bug or a feature, but it is certainly something to be documented (IMO).