Subject: Re: Restrict drag&drop to one level
Posted by mrjt on Thu, 01 Apr 2010 11:22:05 GMT
View Forum Message <> Reply to Message

A bit late replying, but it's an interesting situation and as you say someone else may want the same behaviour.

The DnD stuff is quite elegant but unfortunately quite difficult to understand. We can't all be as smart as Mirek

Basically you have to know that Accept<> and AcceptInternal<> serve a dual function. When the user is dragging (ie on MouseMove) the function does the accept but always returns false so that when you use it in an 'if' statement the actual insert code is avoided. Only when actually doing the 'drop' does it ever return true!

Therefor, to add the level check you must reproduce some of the behaviour from the Accept<> function:

```
 void DropInsert(int parent, int ii, PasteClip& d){
  // Check type of drag data, and restrict to level
  if (IsAvailableInternal<TreeCtrl>(d, "graph") && GetLevel(parent) == 0) {
   // Yes we like this data
   d.Accept();
   // If we haven't dropped the data yet (we are still dragging) don't do anything
   if (!d.IsPaste()) return;
                    // The user has dropped it! Do the insert
   tree.InsertDrop(parent, ii, d);
   tree.SetFocus();
   LOG("accepted graph");
   return;
  }
  if (IsAvailableInternal<TreeCtrl>(d, "data") && GetLevel(parent) == 1) {
   d.Accept();
   if (!d.IsPaste()) return;
   tree.InsertDrop(parent, ii, d);
   tree.SetFocus();
   LOG("accepted data");
   return;
  }
 }
```

You could replace the 'd.Accept' and 'if (d.IsPaste())' with:
if (!AcceptInternal<TreeCtrl>(d, "data")) return;
but it's marginally less efficient.