Subject: Re: Why Splitter::GetChildCount() is not public???
Posted by fudadmin on Tue, 28 Mar 2006 22:45:55 GMT
View Forum Message <> Reply to Message

luzr wrote on Tue, 28 March 2006 22:54fudadmin wrote on Tue, 28 March 2006 16:34luzr wrote on Tue, 28 March 2006 20:31fudadmin wrote on Tue, 28 March 2006 13:18fudadmin wrote on Tue, 28 March 2006 18:11Why Splitter::GetChildCount() is not public???
and PosToClient()...

Well, over time I have found "keep the interface minimal" a good practice. Sometimes I am wrong about interface, but enriching it is quite easy thing. But the opposite, removing stuff from interface, is much more dangerous bussines (angry users... .

Mirek

The users are angry when they find inconsistant interfaces - in some Ctrl's the same methods are public, in some - not. It looks like I will have to maintain my own version of Ultimate++...

PosToClient nor GetChildCount is not common public interface....

Mirek

Originally I meant only count of children. In HeaderCtrl you can have GetCount() which is a number of children? Why all similar controls can't have the same?

Regarding private interfaces - this is where I completely agree with Lisp and Assembler advocates - "C++ was created for stupid programmers"...

P.S
On the other hand Fox-toolkit is not afraid to make public this kind of methods

```
  /// Return true if specified window is owned by this window
  FXbool isOwnerOf(const FXWindow* window) const;

  /// Return true if specified window is ancestor of this window
  FXbool isChildOf(const FXWindow* window) const;

  /// Return true if this window contains child in its subtree
  FXbool containsChild(const FXWindow* child) const;

  /// Return the child window at specified coordinates
  FXWindow* getChildAt(FXint x,FXint y) const;

  /// Return the number of child windows for this window
  FXint numChildren() const;

  /**
```

```
* Return the index (starting from zero) of the specified child window,
* or -1 if the window is not a child or NULL
*/
FXint indexOfChild(const FXWindow *window) const;

/**
* Return the child window at specified index,
* or NULL if the index is negative or out of range
*/
FXWindow* childAtIndex(FXint index) const;

/// Return the common ancestor of window a and window b
static FXWindow* commonAncestor(FXWindow* a,FXWindow* b);
```