Subject: FIXES/ADDS: CtrlLib: properties getters
Posted by kohait00 on Thu, 09 Dec 2010 13:01:49 GMT
View Forum Message <> Reply to Message

hi mirek and others

during my GetProperty/SetProperty devel. i stumbeled over some parameters in some controls, that are good to have getters as well, setters exist but getters are sometimes missing, some const correctnes issues, EditInt64NotNull, EditIntNotNullSpin, EditInt64Spin, EditInt64NotNullSpin, did not exist either.
StringConverter also is missing some getters..
just compare..maybe this can go upstream soon..

EDIT: forgot the archive

File Attachments

1) propchanges.rar, downloaded 300 times

Subject: Re: FIXES/ADDS: CtrlLib: properties getters
Posted by kohait00 on Fri, 10 Dec 2010 08:22:00 GMT
View Forum Message <> Reply to Message

and got some other issue in EditCtrl.h

EditStringNotNull is properly derived from EditString, which is right, since it's the same, except that it comes with NotNull pre-set.

EditIntNotNull, OTOH, is *not* derived from EditInt, but takes a different path with EditMinMaxNotNull as typedef, while EditInt goes the path of EditMinMax as typedef, which makes them difficult to use together, as EditInt, i.e.

EditValue<> <-- EditString <-- EditStringNotNull

/<-- EditMinMax<> == EditInt
EditValue<> -|
\<-- EditMinMaxNotNull<> == EditIntNotNull

but should be like that:

EditValue<> <-- EditMinMax<> == EditInt <-- EditIntNotNull

here come the needed changes for that

EditCtrl.h:260

```
/*
template <class DataType, class Cv>
class EditMinMaxNotNull : public EditValue<DataType, Cv> {
public:
 EditMinMaxNotNull& operator=(const DataType& t)   { EditField::SetData(t); return *this; }

 EditMinMaxNotNull()                          { Cv::NotNull(); }
 EditMinMaxNotNull(DataType min, DataType max)     { Cv::NotNull(); Cv::MinMax(min, max); }

 EditMinMaxNotNull& Min(DataType min)             { Cv::Min(min); Ctrl::Refresh(); return *this; }
 EditMinMaxNotNull& Max(DataType max)              { Cv::Max(max); Ctrl::Refresh(); return *this; }
 EditMinMaxNotNull& NotNull(bool nn = true)       { Cv::NotNull(nn); Ctrl::Refresh(); return *this; }
};
*/

typedef EditMinMax<int, ConvertInt>              EditInt;
typedef EditMinMax<int64, ConvertInt64>          EditInt64;
typedef EditMinMax<double, ConvertDouble>        EditDouble;
typedef EditMinMax<Date, ConvertDate>            EditDate;
typedef EditMinMax<Time, ConvertTime>            EditTime;

/*
typedef EditMinMaxNotNull<int, ConvertInt>       EditIntNotNull;
typedef EditMinMaxNotNull<int64, ConvertInt64>   EditInt64NotNull;
typedef EditMinMaxNotNull<double, ConvertDouble> EditDoubleNotNull;
typedef EditMinMaxNotNull<Date, ConvertDate>     EditDateNotNull;
typedef EditMinMaxNotNull<Time, ConvertTime>     EditTimeNotNull;
*/

class EditIntNotNull : public EditInt
{
public:
 EditIntNotNull& operator=(const String& data)  { SetData(data); return *this; }

 EditIntNotNull() { EditInt::NotNull(); }
 EditIntNotNull(int min, int max) : EditInt(min, max) { EditInt::NotNull(); }
};

class EditInt64NotNull : public EditInt64
{
public:
 EditInt64NotNull& operator=(const String& data)  { SetData(data); return *this; }

 EditInt64NotNull() { EditInt64::NotNull(); }
 EditInt64NotNull(int64 min, int64 max) : EditInt64(min, max) { EditInt64::NotNull(); }
};
```

```
class EditDoubleNotNull : public EditDouble
{
public:
 EditDoubleNotNull& operator=(const String& data)  { SetData(data); return *this; }

 EditDoubleNotNull() { EditDouble::NotNull(); }
 EditDoubleNotNull(double min, double max) : EditDouble(min, max) { EditDouble::NotNull(); }
};

class EditDateNotNull : public EditDate
{
public:
 EditDateNotNull& operator=(const String& data)  { SetData(data); return *this; }

 EditDateNotNull() { EditDate::NotNull(); }
 EditDateNotNull(Date min, Date max) : EditDate(min, max) { EditDate::NotNull(); }
};

class EditTimeNotNull : public EditTime
{
public:
 EditTimeNotNull& operator=(const String& data)  { SetData(data); return *this; }

 EditTimeNotNull() { EditTime::NotNull(); }
 EditTimeNotNull(Time min, Time max) : EditTime(min, max) { EditTime::NotNull(); }
};
```

Subject: Re: FIXES/ADDS: CtrlLib: properties getters
Posted by kohait00 on Wed, 15 Dec 2010 13:40:18 GMT
View Forum Message <> Reply to Message

in another case i stumbled over SpinButtons in EditDoubleSpin (and others), and found out it's private.

what about making it public, to be able to access the Buttons underneath, which are public in SpinButtons as well? so one can use them in a flexible way, hooking up custom Inc Dec operations to them..?

i dont want to start any of those public/private discussions..it's just a little hint, not to be too private..

Subject: Re: FIXES/ADDS: CtrlLib: properties getters
Posted by kohait00 on Tue, 29 Mar 2011 15:46:23 GMT
View Forum Message <> Reply to Message

hi, any comments on that one ?
i keep them around in my tree, hoping to be able to merge again with upstream

EDIT:
attached is a patch file for some of the changes and some usefull fixes, review them please.

File Attachments
1) patch0.patch, downloaded 285 times

Subject: Re: FIXES/ADDS: CtrlLib: properties getters
Posted by mirek on Sun, 10 Apr 2011 09:40:29 GMT
View Forum Message <> Reply to Message

Applied and/or resolved.

I believe that the issue in debug.cpp was addressed by changed LOG definition.

I have renamed the new SpinButtons method to OnSides and provided access to SpinButtons in EditInt/DoubleSpin using method.

I have choosen different approach to the *NotNull hierarchy problem.

Please check.

Subject: Re: FIXES/ADDS: CtrlLib: properties getters
Posted by kohait00 on Sun, 10 Apr 2011 16:04:39 GMT
View Forum Message <> Reply to Message

good work with the NotNull..a lot cleaner than my version.

just a question aside:
some of the things in patch are missing

Pusher

```
bool     IsClickFocus() const { return clickfocus; }
```

SpinButton

```
 bool         IsVisible() const { return visible; }
```

EditCtrl.h / EditField

```
 bool       IsNoBackground() const        { return nobg; }
 bool       IsAlignRight() const          { return alignright; }
```

```
typedef EditMinMaxNotNull<int64, EditInt64>      EditInt64NotNull;
```

```
class EditIntNotNullSpin : public EditIntSpin
{
public:
 EditIntNotNullSpin(int inc = 1) : EditIntSpin(inc) { NotNull(); }
 EditIntNotNullSpin(int min, int max, int inc = 1) : EditIntSpin(min, max, inc) { NotNull(); }
};
```

```
//the EditInt64Spin and EditInt64NotNullSpin

class EditInt64Spin : public EditInt64 {
public:
 typedef EditInt64Spin CLASSNAME;
 virtual void MouseWheel(Point p, int zdelta, dword keyflags);
 virtual bool Key(dword key, int repcnt);

protected:
 SpinButtons sb;
 int64       inc;

 void   Inc();
 void   Dec();
 void   Init();

public:
 EditInt64Spin&  SetInc(int64 _inc)          { inc = _inc; return *this; }
 int64           GetInc() const              { return inc; }
 EditInt64Spin&  OnSides(bool b = true)      { sb.OnSides(b); return *this; }
 bool            IsOnSides() const           { return sb.IsOnSides(); }

 EditInt64Spin&  ShowSpin(bool s = true)     { sb.Show(s); return *this; }
 bool            IsShowSpin() const          { return sb.IsVisible(); }

 SpinButtons&    SpinButtonsObject()         { return sb; }
```

```
 const SpinButtons& SpinButtonsObject() const { return sb; }

 EditInt64Spin(int64 inc = 1);
 EditInt64Spin(int64 min, int64 max, int64 inc = 1);
 virtual ~EditInt64Spin();
};

class EditInt64NotNullSpin : public EditInt64Spin
{
public:
 EditInt64NotNullSpin(int inc = 1) : EditInt64Spin(inc) { NotNull(); }
 EditInt64NotNullSpin(int min, int max, int inc = 1) : EditInt64Spin(min, max, inc) { NotNull(); }
};
```

see here the

```
 bool          IsShowSpin() const          { return sb.IsVisible(); }
```

of the SpinButton exposure, as opposed to ShowSpin() in the Spin variants

BTW: why not have the

```
 EditInt64Spin(int64 inc = 1);
 EditInt64Spin(int64 min, int64 max, int64 inc = 1);
```

for the EditCtrl changes, see the attached files, just replace..

i think this one hasnt been in the patch..
ProgressInfo

```
 int   Get() const                  { return actual; }
 int   GetTotal() const               { return total; }
```

File Attachments

1) EditCtrl.h, downloaded 379 times
2) EditField.cpp, downloaded 290 times

Subject: Re: FIXES/ADDS: CtrlLib: properties getters
Posted by mirek on Sat, 16 Apr 2011 17:58:25 GMT
View Forum Message <> Reply to Message

Applied. Sorry for forgetting some of them before; to my defence, that patch format was not recognized by any of my tools so I have to apply it manually.

Mirek

Subject: Re: FIXES/ADDS: CtrlLib: properties getters
Posted by chickenk on Sat, 16 Apr 2011 18:37:53 GMT
View Forum Message <> Reply to Message

mirek wrote on Sat, 16 April 2011 19:58Applied. Sorry for forgetting some of them before; to my defence, that patch format was not recognized by any of my tools so I have to apply it manually.

Mirek

A git-formatted patch? I'm very surprised because it's a very standard unified diff... Maybe because of the 'a/' and 'b/' prefixes to folders...

Did you try with 'patch -p1 < patch0.patch' in 'trunk'? It should have worked...

Subject: Re: FIXES/ADDS: CtrlLib: properties getters
Posted by mirek on Sat, 16 Apr 2011 18:45:31 GMT
View Forum Message <> Reply to Message

chickenk wrote on Sat, 16 April 2011 14:37mirek wrote on Sat, 16 April 2011 19:58Applied. Sorry for forgetting some of them before; to my defence, that patch format was not recognized by any of my tools so I have to apply it manually.

Mirek

A git-formatted patch? I'm very surprised because it's a very standard unified diff... Maybe because of the 'a/' and 'b/' prefixes to folders...

Did you try with 'patch -p1 < patch0.patch' in 'trunk'? It should have worked...

I am not in posix environment doing this. Plus, patch is not good enough for me to do this work. I need to review all changes manually.

Right now, I am using tortoise-svn to apply patches. It did not understood this format.

Subject: Re: FIXES/ADDS: CtrlLib: properties getters
Posted by kohait00 on Sun, 17 Apr 2011 12:37:07 GMT
View Forum Message <> Reply to Message

whats the best format for you to handle stuff?

(i dare to vote for GIT again, where you simply can, after reviewing the changes, cherry-pick the

commits from some fellows or make your own changes, or make patches, or apply them.. or..

thanks btw for taking those patches.

Subject: Re: FIXES/ADDS: CtrlLib: properties getters
Posted by kohait00 on Tue, 19 Apr 2011 14:31:13 GMT
View Forum Message <> Reply to Message

one more is pending..
Then my CtrlPos package works again with upstream

Progress.h:80

```
int   Get() const                { return actual; }
int   GetTotal() const             { return total; }
```

Subject: Re: FIXES/ADDS: CtrlLib: properties getters
Posted by mirek on Tue, 19 Apr 2011 16:56:46 GMT
View Forum Message <> Reply to Message

That is weird, I see Progress already has GetPos and GetTotal and moreover there is no 'actual' member variable...

Subject: Re: FIXES/ADDS: CtrlLib: properties getters
Posted by kohait00 on Tue, 19 Apr 2011 18:11:45 GMT
View Forum Message <> Reply to Message

you're right.
i'm sorry, it's about ProgressIndicator.
swapped the lines somehow it's line 25..

Subject: Re: FIXES/ADDS: CtrlLib: properties getters
Posted by mirek on Fri, 22 Apr 2011 12:41:19 GMT
View Forum Message <> Reply to Message

Applied.
