

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::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(); }  
};
```
