Subject: int <-> String conversions
Posted by [hojtsy](#) on Sun, 22 Jan 2006 13:41:23 GMT

I created a small dummy application with a StaticText and a Button. The Button is supposed to increment the int value displayed in the StaticText.

```
MyApp::MyApp()
{
  CtrlLayout(*this, "Window title");
  plus <<= THISBACK(onPlus);
  text.SetText("0");
  Zoomable().Sizeable();
}

void MyApp::onPlus()
{
  int val = StdConvertInt().Scan(text.GetText());
  text.SetText(StdConvertInt().Format(val + 1).ToString());
}
```
Is there a more simple way to do what onPlus does? It seems ugly.

Subject: Re: int <-> String conversions
Posted by [zsolt](#) on Sun, 22 Jan 2006 15:13:04 GMT

Try using EditInt or EditIntSpin.

Subject: Re: int <-> String conversions
Posted by [hojtsy](#) on Sun, 22 Jan 2006 15:17:10 GMT

I know those widgets. This is just a dummy example for int <-> String conversion in u++. Do you know an easier way for that?

Subject: Re: int <-> String conversions
Posted by [mirek](#) on Mon, 23 Jan 2006 15:28:57 GMT

int->String:

AsString(x)

BTW, AsString is the "standard" function, it is implemented for many types. Moreover, templated

---

operator<<(String&, T)

and

operator<<(Stream&, T)

are implemented to use AsString, so in most cases, when putting things together, you do not need to care and just put any type on the right side of "<<":

String r; // or e.g. FileOut r;
int i;
Date d;
r << "The number is: " << i << " and the date is: " << d;

String->int:

Well, you can use c library atoi for simple cases. If you are about to have Null option for the empty string, you can use StrIntValue function.

However, in most cases, this is the part of wider text parsing code and that domain is completely ruled by CParser - simple, but extremely useful lexical parser.