

---

Subject: Core split

Posted by [arturbac](#) on Fri, 03 Aug 2007 15:32:37 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

It would be nice to have all tempalte container classes separated in different headers.

Concrete implementation of Core makes a bit difficult using NTL with other toolkits.

And it is completly not required to use all things from Core if I just want to use NTL Vector and Array.

For now that is imposible because Vcont.h even dont have #pragma once.

Core would be much better with STL,Bost, Agg idea of separating algorithms into separate and independant code.

This is just Idea after Gdiplus , Core type colissions ...

I connt use

```
use namespace Gdiplus;
```

```
use namespace Upp;
```

I have to type by hand all things from one of library ie.

```
using namespace Upp;
```

```
using Gdiplus::Pen;
```

```
typedef Gdiplus::Color Color_;
```

```
very long list follows.....
```

It is just Idea .. (not complaining)

---

---

Subject: Re: Core split

Posted by [mirek](#) on Fri, 03 Aug 2007 16:56:47 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Years ago, I have separated NTL from U++ as standalone library. Interest was close to zero.

Also, there is fragile problem with U++ maps - they require moveable keys. With U++, it is simple.

In fact, most existing std::string as well as CString implementations are moveable too, but you cannot take it granted...

---

---

Subject: Re: Core split

Posted by [arturbac](#) on Fri, 03 Aug 2007 21:02:04 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

First thing i do was to disable MoveableAssert ...

Decalring NTL\_MOVEABLE(type) in header causes error ...

---

---

Subject: Re: Core split

Posted by [mirek](#) on Sat, 04 Aug 2007 07:49:16 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

arturbac wrote on Fri, 03 August 2007 17:02First thing i do was to disable MoveableAssert ...

Decalring NTL\_MOVEABLE(type) in header causes error ...

Well, that is OK as long as types are really moveable..

---