Subject: Re: why no virtual destructor for RegExp?
Posted by navi on Mon, 28 Jan 2013 06:11:31 GMT
View Forum Message <> Reply to Message

DONE.

Final Interface. Got rid of pattern and offset. 3 flavors of each. Also updated the demo, see updated screen cap above.

```
int  Replace(String& t, Vector<String>& rv, bool backref=false); //DONE
int  Replace(String& t, const String r, bool backref=false); //DONE
int  Replace(String& t, Callback1<Vector<String>&> cbr); //DONE

int  ReplaceGlobal(String& t, Vector<String>& rv, bool backref=false); //DONE
int  ReplaceGlobal(String& t, const String r, bool backref=false); //DONE
int  ReplaceGlobal(String& t, Callback1<Vector<String>&> cbr); //DONE
```

Callback works in same fashion as in U++ Menu and Bar. User writes a callback function with void return and "Vector<String>& v" argument. And uses the THISBACK macro to pass it to Replace() or ReplaceGlobal(). Inside the match_callback_fun() user gets a vector already populated with all the matches. any changes made to the vector will be reflected in the source string 't'.

```
void match_callback_fun(Vector<String>& v){
...
}

regx.Replace(t, THISBACK(match_callback_fun));
regx.ReplaceGlobal(t, THISBACK(match_callback_fun));
```

Regards
Navi

File Attachments

1) RegExp Extension 28-01-2013 #1700.rar, downloaded 370 times