
Subject: Re: RE: Job package: A scope-bound worker thread for non-blocking operations.

Posted by [mirek](#) on Tue, 19 Sep 2017 06:37:02 GMT

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

This is ugly:

```
// Reuse workers and make requests in a simple, non-blocking way.
{
  Cout() << "----- Processing individual non-blocking requests...\n";
  // We can "clear" the data. Useful especially if the data is "picked".
  c1.Clear();
  c2.Clear();

  c1.Blocking(false).Request(host1, 21);
  c2.Blocking(false).Request(host2, 21);

  while(!c1.IsFinished() || !c2.IsFinished())
    ;
  if(c1.IsError()) Cerr() << c1.GetErrorDesc() << '\n';
  else Cout() << ~c1 << '\n';

  if(c2.IsError()) Cerr() << c2.GetErrorDesc() << '\n';
  else Cout() << ~c2 << '\n';

}
```

I think this should work without IsFinished loop.

Anyway, all thing still feels very much like future, with implicit promise connected to Thread. In that regard, full future/promise still sounds more powerful.

Mirek
