Subject: Re: RE: Job package: A scope-bound worker thread for non-blocking operations.
Posted by mirek on Wed, 11 Oct 2017 17:30:58 GMT
View Forum Message <> Reply to Message

Weird. The only reason for erratic behaviour would be some form of race condition, so I have tried this:

```
#include <Core/Core.h>

using namespace Upp;

Atomic h;

void Wait(int c)
{
 while(c--)
  h++;
}

CONSOLE_APP_MAIN
{
 StdLogSetup(LOG_COUT|LOG_FILE);

 for(int i = 0; i < 100000; i++) {
  if(i % 1000 == 0)
   LOG(i);
  bool b = false;
  auto x = Async([] { Wait(Random(3000)); throw "error"; });
  try {
   Wait(Random(3000));
   x.Get();
  }
  catch(...) {
   b = true;
  }
  ASSERT(b);
 }

 LOG("=========== OK");
}
```

... and it passed without a problem. Added that test to autotest just to be sure...

Anyway, (maybe I forgot to mention to AsyncWork is now in U++): Are you trying with current trunk?