Subject: Re: XML Rpc client will halt when server is not running
Posted by kasome on Thu, 21 Jun 2012 03:21:31 GMT
View Forum Message <> Reply to Message

Here is full XML RPC Server & Client Code

Server:

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

using namespace Upp;

XMLRPC_METHOD( Compute ) {
 double a, b;
 Upp::String arithmeticOperator;
 rpc >> a >> arithmeticOperator >> b;
 LOG( Upp::Format("Request: %nf %s %nf", a, arithmeticOperator, b) );

 if( arithmeticOperator.GetCount() == 1 ) {
  switch( *arithmeticOperator ) {
   case '+': {
    rpc << a + b;
    break;
   }
   case '-': {
    rpc << a - b;
    break;
   }
   case '/': {
    if( b == 0 ) {
     rpc << Upp::ErrorValue("division by zero");
    }
    else {
     rpc << a / b;
    }
    break;
   }
   case '*': {
    rpc << a * b;
    break;
   }
  }
 }
 else {
  rpc << Upp::ErrorValue("unknown operator");
 }
```

```
}

XMLRPC_METHOD( GetServerTime ) {
 LOG( "Request: GetServerTime" );
 rpc << Upp::GetSysTime();
}

int main() {
 TcpSocket rpc;
 int port = 1234;

 if( !rpc.Listen(port,5) ) {
  return false;
 }

 while( true ) {
  if( _kbhit() ) {
   if( _getch() == 27 ){
    break;
   }
  }

  TcpSocket http;
  http.Timeout(1000);
  if( http.Accept(rpc) ) {
   XmlRpcPerform(http,NULL);
  }
 }

 return 0;
}
```

Client:

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

using namespace Upp;

namespace Upp {
 extern bool HttpRequest_Trace__;
}

void Compute( double a, Upp::String arithmeticOperator, double b );

int main() {
```

```
 Time serverTime;
 SetDateFormat( "%4d-%02d-%02d" );
 XmlRpcRequest call( "127.0.0.1:1234" );
 if( call("GetServerTime") >> serverTime ) {
  LOG( "Server Time = " << serverTime );
 }
 else {
  LOG( Upp::Format("Error: %s", call.GetError()) );
 }

 Compute( 12, "+", 12 );
 Compute( 12, "*", 12 );
 Compute( 12, "+56", 12 );
 Compute( 12, "/", 0 );

 return 0;
}

void Compute( double a, Upp::String arithmeticOperator, double b ) {
 double result = 0;
 XmlRpcRequest call( "127.0.0.1:1234" );
 if( call("Compute", a, arithmeticOperator, b) >> result ) {
  LOG( Upp::Format("%f %s %f = %f", a, arithmeticOperator, b, result) );
 }
 else {
  LOG( Upp::Format("Error: %s", call.GetError()) );
 }
}
```