

## Quick Setup to Remote Debugging in MS VC++ 6

By:  
Michaël Schoonbrood

September 4<sup>th</sup> 2001  
Version 1.0  
Best viewed with Microsoft Word for Windows 97 or better.

---

Web Page:  
<http://unitek3000.tripod.com/>

---

Any Doubts?  
[unite\\_future@hotmail.com](mailto:unite_future@hotmail.com)

---

Length of Tutorial:  
You should take half an hour off to complete this tutorial.

---

Necessary Software:  
You need MS Visual C++ 6, 2 PC's and a LAN to complete this tutorial.

---

Files:  
This tutorial has no accompanying files.

---

Message:  
These Tutorials take time to make. The authors took all the necessary steps to ensure the Tutorials are accurate and fault-free.  
Even so, once in a while, errors get through. It is, in my opinion, unavoidable.  
So, to all of you reading this, in case you detect an error in the tutorial, please email us so we can correct it.  
Also, be sure to email the authors with your doubts and questions. These questions will enable them to write better and better tutorials.  
And, any suggestion you have for us... well, you know where to reach us.

The team thanks your cooperation.

The best way to debug your full screen games (and other applications) is by remote debugging. Remote debugging means that you run the EXE on PC2, while you debug on PC 1. Very cool... This page contains a textual description, the following page contains a quick setup guide which you can use to quickly setup (or verify your setup) your remote debugging settings.

Why is it better than dual monitor debugging?

When using dual monitor to debug your application, you still share the keyboard between your development environment and your running EXE. Also, when debugging messages, you're out of luck in a lot of cases, because the activation of the debug environment (breakpoint) generates FOCUS\_LOST messages and all other unwanted messy stuff.

Here's the hardware you need:

2 PC's connected through a LAN (I'm not gonna do a tutorial on how to setup a home LAN... they are all over the web!). MS Visual Studio needs to be installed on both machines.

How does it work?

The easiest way to get it working is to share your projects debug dir. The other PC needs to map a drive letter to this debug dir! You need this drive letter on PC 1 later on, to set the debug executable.

Ok.. here we go..

Open MSVC, and start a new MFCAppWizard project called "RemoteDebugTest". Click on [DialogBased] and [Finish].

Open the Windows Explorer and share your debug dir (in my case "C:\Work\RemoteDebugTest\Debug") under the share name "DEBUG".

Go to your other PC and map drive M to the newly shared DEBUG dir.

While your behind system 1, run MSVCMON.EXE (it should be in your MSDev98\Bin dir, otherwise search for it). Click on settings. Targetmachine is the name or IP of the PC your debugging on. Click OK and back on the main screen click connect. This system is now ready for remote debugging.

Back on PC1 (where you created your project in MSVC) click on the BUILD menu and select [Debugger remote connection]. Set connection to [Network TCP/IP]. Next click on settings and at targetname enter the name or IP of the system that's now running MSVCMON and awaiting your connection. Click [ok] and [ok] (you should be back in the workspace).

Almost done.. hang in there...

Go to [Project] in the menu and select [Settings]. Go to the debug tab and set remote debug to "M:\RemoteDebugTest.exe".

Compile and run... voila!

Quick setup guide:

Here is my setup:

Hostmachine ("Workstation" IP 192.168.0.1 SUBNET 255.255.255.0):

Workdir = C:\WORK\PROJECTNAME

Extra directory share = C:\WORK\PROJECTNAME\DEBUG

share name = DEBUG

In MSVC6 menu: Build -> Debugger Remote Connection

Connection = Network TCP/IP

Settings, Target machine = Debugstation

In MSVC6 menu: Project -> Settings

make sure you select the project name in the tree (left side)

on the DEBUG tab, set remote debug to = M:\PROJECTNAME.exe

Debugmachine ("Debugstation" IP 192.168.0.2 SUBNET 255.255.255.0):

Drive mapping (drive M) to \\Workstation\DEBUG

Run MSVCMON.EXE

Connection = Network TCP/IP

Settings, Target machine = Workstation

Click connect...

it will now start waiting for your debug

You're ready to go!