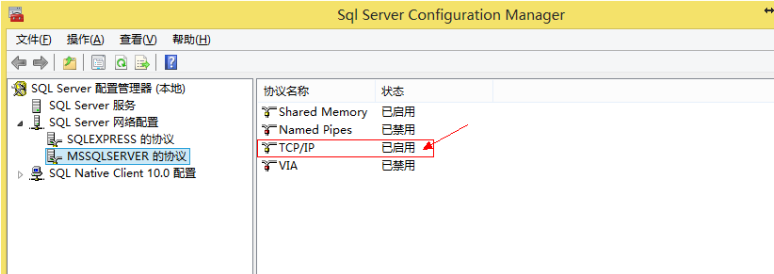
TCP/IP的基本工作原理这里就不浪费口水了。现在这网络年代，谁不知道TCP/IP啊。不要跟我抬杠说你偏就没听说过TCP/IP阿，真是这样那你得自己去补补课了。

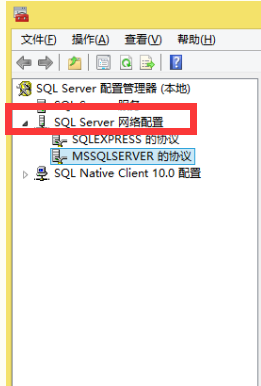
TCP/IP协议有两个基本的东西，一个是IP地址， 另一个是端口号。 在SQL Server 上使用TCP/IP协议是非常简单的。首先要对SQL Server所在机器的网卡配置好TCP/IP协议并得到一个IP地址（通过静态指定分配或从DHCP服务器动态获得都行），接下来就需要通过SQL Server提供的工具做服务器端设置。

**一、Microsoft SQL Server 上的TCP/IP协议**

在运行SQL Server 2000的服务器上，运行服务器端网络配置工具（运行svrnetcn.exe即可调出）配置SQL Server 监听TCP/IP协议即可，如图所示：



在运行Microsoft SQL Server 2005/2008 上的服务器上，配置TCP/IP协议需要在SQL Server配置管理器(Configuration Manager)工具下的Network Configuration下面。下面是这个工具的截屏。



其实，不论是SQL Server 2000还是SQL Server 2005/2008,配置的结果都是存放在注册表HKEY\_LOCAL\_MACHINE \SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.\MSSQLServer\SuperSocketNetLib下面的各个项目里。不用管理界面，直接修改注册表也能达到一样的目的。

（要注意，如果你的机器上只有SQL Server 2000的默认实例，那么注册表所在位置会稍有不同，它是HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\MSSQLServer\MSSQLServer\SuperSocketNetLib.）

配置好网络协议以后，需要重新启动SQL Server服务，让修改生效。启动后，需要检查SQL Server的errorlog，确认这些协议都已经被正常开启。

Shared Memory正常启动，可以看到如下信息。

2009-04-12 10:04:27.92 Server      **Server local connection provider** is ready to accept connection on [ \\.\pipe\SQLLocal\MSSQLSERVER ].

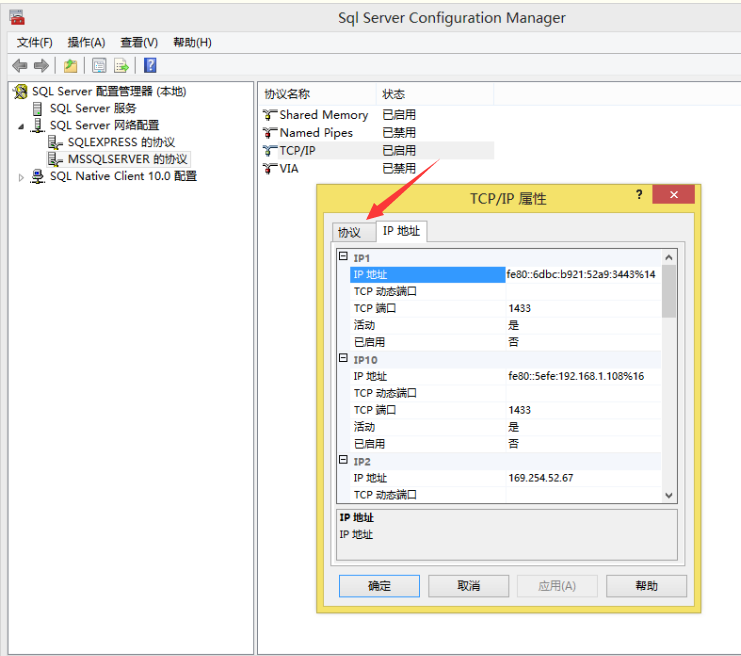
Named Pipe正常启动，可以看到如下信息。

2009-04-12 10:04:27.92 Server      **Server named pipe provider** is ready to accept connection on [ \\.\pipe\sql\query ].

TCP/IP正常启动，可以看到SQL Server实例正在侦听的IP地址和Port号。例如：

2009-04-12 10:04:27.92 Server      Server is listening on [ 'any' 1433].

--侦听服务器上所有IP地址上的1433端口（将协议中的侦听所有设置为是，并在IP地址栏中最后的配置IPALL设置为1433接口，然后重启服务）。



或者：

2008-06-12 15:01:58.150 Server       Server is listening on [ 172.30.30.80 1433].

--只侦听指定的IP地址上（172.30.30.80）的1433端口（需要在管理器的TCP/IP的协议中:1、将侦听所有设置为否，

2、然后在IP地址的指定ip上将Enable或者启用设置为true，

3、然后将TCP动态端口留空不要写任何字符，

4、最后将TCP端口指定为1433

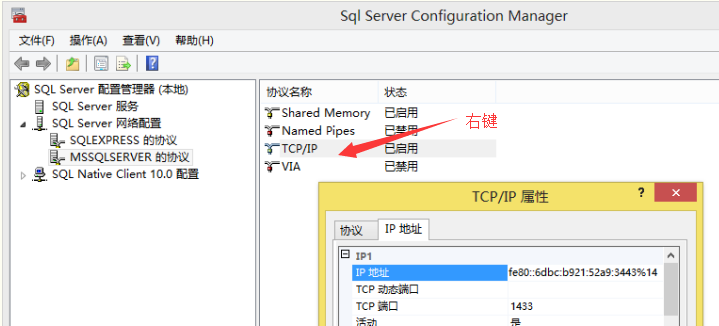
5、设置好后应用并确定然后重启服务）。

这里也顺便回答一个小问题。 有些朋友问， 如果机器的IP地址改变，对SQL Server有没有影响， 需要作什么动作等。答案很简单， 重新启动SQL Server 服务就可以了。 SQL Server 会自动监听机器的新IP地址，无需重新安装或配置SQL Server。

**二、Microsoft SQL Server 2000 监听的TCP/IP端口号**

现在谈谈一个重要的配置即端口号。在服务器端网络配置工具中或者在配置管理器中选中TCP/IP协议，点击Properties（属性）按钮，就可以查看SQL Server侦听的端口号：

 （SQL Server 2005/2008配置管理器）



这里显示的就是SQL Server监听的 TCP/IP 端口。默认值为 1433。SQL Server服务器将在该端口上监听并接受客户端的TCP/IP socket连接请求。你可以改变这个默认值为别的端口号，只要该端口没有被系统或别的应用程序占用即可。一般说来高于5000以上的端口号码都可以随意使用，或者使用小于1024的操作系统或其他应用程序不使用的空闲端口。为什么不用1024至5000之间的端口呢？ 因为缺省情况下操作系统会把这个范围内的端口分配给服务或应用程序使用。为了避免潜在的端口冲突所以使用5000以上或1024以下的值。实际上不论那个范围，只需要是空闲的没使用的端口都可以。微软有一篇很好的技术文档详细讨论了Windows系统使用一些端口号，文档连接如下：

Microsoft Windows 服务器系统的端口要求

<http://support.microsoft.com/?id=832017>

或许你会问，有没有方法查看SQL Server 监听的端口啊？有的，可以使用netstat命令。 在DOS窗口下运行 “netstat -an”命令即可列出系统所有使用中的端口号，当然SQL Server 监听的端口也在其中了。下面是从“netstat -an”命令的输出中摘取的有关SQL Server的端口号部分：

Active Connections

  Proto  Local Address          Foreign Address        State

  TCP    0.0.0.0:1433           0.0.0.0:0              LISTENING

  TCP    169.254.173.244:1433   169.254.173.244:3952   ESTABLISHED

UDP    0.0.0.0:1434           \*:\*

上面列出的1434端口是比较有趣的一个端口，详见后面的内容。

SQL Server支持一台机器安装多套实例，即默认实例和命名实例。对于默认实例，缺省侦听的IP端口是1433。如果你把默认实例的监听端口改为非1433端口， 那么客户端就有可能连接不上SQL server默认实例。 这个时候必须使用客户端网络工具指定SQL Server 侦听的IP端口或创建一个指定IP端口的SQL Server别名（指定别名时需要启动sql server browser服务，该服务会监听端口为1434，会接收要连接客户端发送的带别名的upd数据连接包，该服务会枚举所有的实例，然后将匹配实例名对应的实际ip和端口返回给客户端，客户端在以此ip和端口访问特定的实例）。关于客户端网络工具在下面会具体讲。

**三、TCP/IP静态端口，动态端口，SQL Server Browser和UDP 1434端口的概念**

首先讨论1434 端口。1434端口是用来干吗的呢？

我们都知道 SQL Server 支持多个实例，缺省的 SQL Server 实例使用默认的1433端口。对于其余的命名实例(Named Instance），每次启动所绑定的端口号可能都不一样。让每个用户都去记住SQL的端口号是不可能的。一个数据库普通用户只会知道数据库服务器的名字和实例名。那么怎么根据服务器名和实例名找到相应的端口号呢？SQL Server 2000的时代，产品组开发了一套SQL Server 解析协议 (SSRP)，用于侦听 UDP 1434 端口。当 SQL Server 2000 客户端 Net-Library 连接到 SQL Server 2000 实例时，仅需要运行该实例的计算机所在的网络名和该实例名（如MYSQLSrv\inst1）。当应用程序请求连接SQL Server服务器时，客户端网络库发送一个网络包到SQL Server 服务器的1434端口（udp协议）。所有运行 SQL Server 2000 实例的计算机都监听此端口。服务器收到查询网络包时将返回一个包含服务器上运行的所有实例监听信息的数据包。对于每个实例，该数据包报告该实例正在监听的服务器 Net-Library 和具体网络地址。客户端应用程序根据返回的网络地址进行连接。 举个具体的例子，对于命名实例，该返回的网络包中包含了SQL Server 服务器正在监听的TCP/IP端口，客户端应用程序得到端口号后才能和SQL Server 建立连接（相比直接使用ip+端口号连接会慢一些，upd的不安全性也可能到会连接不上）。

这个设计理念本身是挺好的。但是在2003年，一个叫Slammer的病毒利用SSRP功能组件里的一个bug，诱导SQL Server服务在UDP端口发出大量网络包，从而导致了网络阻塞，并且使数据库服务瘫痪的严重后果。这个病毒是和SQL Server相关的迄今为止危害最大的病毒。为了避免这类悲剧事件再次发生，SQL Server 2005 引入了 SQL Server Browser 服务来替换原有的机制。

SQL Server browser用 SQL Server 解析协议 (SSRP) 侦听 UDP 端口，并接受未经身份验证的请求。为了防止恶意用户利用这个服务攻击SQL Server服务器，SQL Server 浏览器可以设置在低特权用户的安全上下文中运行，将受到恶意攻击的几率降到最低。可以将一个新建用户加入SQLServerXXXXSQLBrowser$这个本地组里。并且需要通过使用 SQL Server 配置管理器更改登录帐户。SQL Server 浏览器的最小用户权限如下：

* 拒绝通过网络访问该计算机
* 拒绝本地登录
* 拒绝以批处理作业登录
* 拒绝通过“终端服务”登录
* 作为服务登录
* 读取和写入与网络通信（端口和管道）相关的 SQL Server 注册表项

通过这样的设计，可以隔离恶意网络攻击对SQL Server服务本身的影响，并且通过限制SQL Browser登录用户权限的方法，限制了万一SQL Browser受到攻击，对整个服务器的影响。

启动SQL Server Browser后，它将启动并使用 UDP 1434 端口。SQL Server 浏览器将读取注册表（所有HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.X\MSSQLServer\SuperSocketNetLib下的记录），识别计算机上的所有 SQL Server 实例，并注明它们使用的端口和命名管道。当一台服务器具有两个或多个网卡时，SQL Server 浏览器会为 SQL Server 返回其遇到的第一个已启用的端口。

客户端使用实例名连接服务器流程：当 SQL Server 客户端请求 SQL Server 资源时，客户端数据驱动程序将使用 1434 端口向服务器发送一条 UDP 消息。SQL Server 浏览器将会响ss应请求实例的 TCP/IP 端口或命名管道名称。然后，客户端数据驱动程序将使用所需实例的端口或命名管道向服务器发送请求来完成连接。

当 SQL Server Browser 服务不运行时，如果您提供了正确的端口号或命名管道，仍可以连接到 SQL Server。如果 SQL Server 的默认实例在 1433 端口上运行，则可以使用 TCP/IP 连接到此默认实例。

但是，如果 SQL Server Browser 服务未运行，则以下连接无效：

* 在未完全指定所有参数（例如 TCP/IP 端口或命名管道）的情况下，组件尝试连接到命名实例。
* 生成或传递其他组件随后要用来进行重新连接的服务器/实例信息的组件。
* 未提供端口号或管道就连接到命名实例。
* 在未使用 TCP/IP 1433 端口的情况下，将 DAC 连接到命名实例或默认实例。
* 枚举 SQL Server Management Studio、企业管理器或查询分析器中的服务器。

如果应用程序通过网络访问 SQL Server，若要停止或禁用 SQL Server Browser 服务，必须为每个实例分配一个特定端口号，并在客户端应用程序代码中指定该端口号（固定的端口号而非动态的，动态的端口号和实例名能够被sql server browser记录，而停止sql server browser服务时则端口号不是确定的，所有在停止sql server browser服务时要为各个实例绑定固定的端口号）。但此方法存在如下问题：

* 必须更新和维护客户端应用程序代码才能确保它连接到正确的端口。
* 如果服务器上的其他服务或应用程序占用了您为每个实例选择的端口，则会导致 SQL Server 实例不可用。

所以SQL Browser这个服务做的事情虽然很简单，但对正常的客户端连接非常重要。如果某些客户端连不上SQL Server，报告“SQL Server doesn't exist or access denied”，可以尝试指定端口，看看能不能连上。如果这样能够连上，一般是因为UDP 1434在网络上被禁用了，需要在防火墙或者网关上打开这个端口。

SQL Browser本身很少出现问题。不过由于其设计比较简单，一共只有4个线程在接受SSRP的包（2个给IPv4，2个给IPv6）。因此一旦这些线程都因为异常而终止的话，可能会出现SQL Browser服务运行正常但是连接命名实例有问题的情况。这个时候你会在Windows的事件日志里看到以下错误：

The SQLBrowser processing of requests against a particular IP address has encountered a critical error. Processing of requests on this address has been halted (event ID 14)

微软已经有发布了一个补丁程序来解决这个问题，参见：

<http://support.microsoft.com/kb/2526552>

另外需要注意的是SQL Browser启动账号要有 读取和写入与网络通信（端口和管道）相关的 SQL Server 注册表项（HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.X\MSSQLServer\SuperSocketNetLib）的权力。如果注册表读不到，SQL Browser不会报错，但是就不告诉客户端想要的port和pipe的信息，也会导致客户端连不上。

再来讨论静态端口。本文一开头提及如何修改SQL Server 监听的端口。 如果你配置的端口是非0值那么就是静态端口了。也即是我们指定了一个静态的端口号，SQL Server 将在该端口上监听。这对于默认实例和命名实例都是一样的。只是对于默认实例，如果端口不是1433，那么得在客户端使用网络配置工具指定缺省连接端口（指定不带端口时连接那个预定义端口）或创建SQL别名来进行连接。而对于命名实例则不一样。你可以指定任意端口值。只要SQL Server 在该端口进行监听，那么客户端就能够通过1434端口查询出该命名实例的端口值而进行连接（需要sql server browser 服务启动运行并且没有被防火墙限制）。

另外，万一指定的端口已经被占用时，SQL Server 又会怎么办呢？ 对于默认实例，SQL Server 干脆就放弃监听TCP/IP协议并在日志中记录“不能监听TCP端口”类似的信息。而对于命名实例则聪明一些。SQL Server 命名实例会自动选择下一个空闲的端口来监听。

如果把SQL Server 监听的端口设为0又会如何？这个问题比较有趣。事实是，当设为0时，无论是默认实例还是命名实例，SQL server 都理解为需要动态配置监听的端口。动态配置的意思是，它们都会自动选择一个系统空闲的端口进行监听。但记住，对于默认实例，如果选择的端口不是1433的话，嘿嘿，你得需要在客户端使用cliconfg.exe工具配置缺省连接端口或使用别名了。

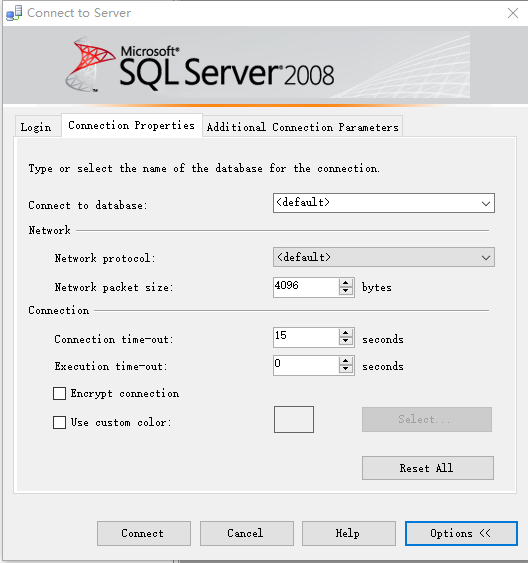
动态配置端口是否意味着SQL Server 每次重新启动后都会随机选择一个未用端口呢？这倒不是的。 当重新启动后，SQL Server 将设法侦听之前使用的端口（这也是为什么配置完动态端口后多次重启sql server服务监听的端口号都没有改变）。如果 SQL Server 不能绑定到该端口，那么会动态地绑定到另一个空闲端口（所以在使用动态端口而不开启sql server browser服务的时候最好指定静态端口）。

**四。客户端的TCP/IP协议配置**

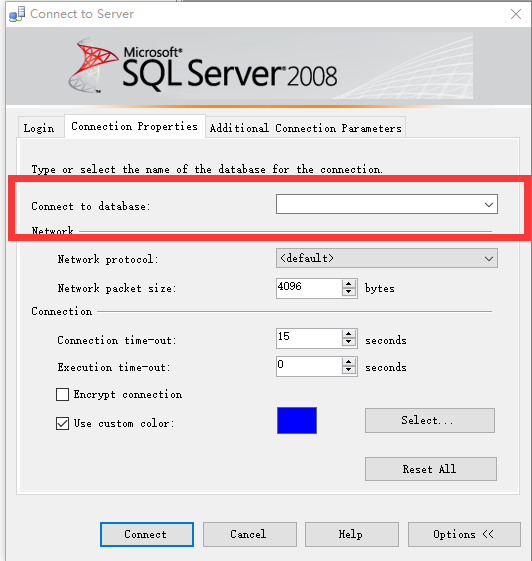
大多数情况下在客户端你不需要进行配置，因为TCP/IP协议默认是启用的。和命名管道一样，可以使用客户端网络实用工具进行配置TCP/IP协议。客户端应用程序都是通过加载SQL Server的数据驱动控件做SQL Server连接的。现在客户端数据驱动库主要有2种：

1．MDAC (Microsoft 数据访问组件)

运行cliconfg.exe或从开始-->程序-->Microsoft SQL Server -->客户端网络实用工具即可调出。



客户端网络实用工具中左边是禁用的协议，右边的是启用的协议。如果右边没有TCP/IP协议，则点击Enable按钮启用它。 选中TCP/IP后点击属性（Properties）按钮即可查看客户端连接的缺省的TCP/IP端口。缺省情况下1433（参考图4）。如果服务器默认实例（不是命名实例）监听的端口不是1433，那么你需要在这里把default port作相应的改变。当然你可以像在命名管道篇中介绍的那样创建服务器别名特别指定服务器的端口也可以。对于命名实例， 客户端网络库会利用UDP 1434端口查询服务器命名实例的监听协议信息。



可以在客户端配置SQL Server别名，以明确指定连接到SQL Server所使用的协议。注意在别名中可以指定端口（当指定端口的时候会有先使用端口而话别名，所有可能导致实际访问的不是别名指定的实例而是监听该端口的实例），也可以使用动态查询端口功能。

当默认实例被配置为侦听一个非1433端口我们就可以通过配置默认端口或者别名来让客户端程序找到默认实例。

配置的信息保存在注册表里，HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\SuperSocketNetLib子目录下面。也可以直接改值，一样能达到效果。

2. SQL Server Native Client

在一台没有安装过SQL Server 2005或者2008的机器上，缺省不会安装SQL Server Native Client。在安装 SQL Server 2008 或 SQL Server 客户端工具时，将同时安装 Microsoft SQL Server Native Client 10.0。如果计算机上还安装了 SQL Server Native Client 的 SQL Server 2005 版本，则 SQL Server Native Client 10.0 将与早期版本Microsoft SQL Server Native Client 9.0并行安装。

如果安装有SQL Server 客户端工具，我们可以通过SQL Server Configuration Manager来配置客户端网络协议。配置的方法和MDAC类似。

图 5 - 5

如果没有安装这个工具，可能就需要直接修改注册表了。Microsoft SQL Server Native Client 9.0的信息存放在HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\SNI9.0下面，Microsoft SQL Server Native Client 10.0的信息存放在HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\SNI10.0下面。

**五。 TCP/IP连接问题的解决步骤**

**步骤1: 验证SQL Server 是否真的监听了TCP/IP协议**

为了验证SQL Server 确实监听了TCP/IP协议，可以打开SQL Server Query Analyzer （查询分析器），然后运行如下命令：

exec master..xp\_readerrorlog

在结果栏，如果看到类似如下一行则表明SQL Server已经监听了TCP/IP：

2000-08-31 21:47:01.52 server    SQL server listening on 169.254.173.244: 1433.

2000-08-31 21:47:01.52 server    SQL server listening on 127.0.0.1: 1433.

2000-08-31 21:47:01.53 server    SQL server listening on TCP, Shared Memory.

如果发现SQL Server 没有监听TCP/IP协议，请使用服务器端网络配置工具（运行svrnetcn.exe即可调出）确认是否配置好SQL Server 监听TCP/IP协议。

**步骤2: 验证服务器监听的TCP/IP端口和客户端配置的缺省值或别名中指定的值一致。**

使用客户端网络实用工具检查客户端的连接协议配置，确保客户端启用了TCP/IP。当然，客户端连接的缺省端口需要和SQL服务器监听的一致。另外，如果有别名，需要仔细查看其指定的端口是否正确。如果客户端的别名设置错误，也会引起连接问题。

**步骤3:检查网络连通性。**

要确保不但能够ping通 SQL Server服务器的IP地址，也能够ping 通SQL Server服务器的名称。如果ping 服务器名字有问题， 说明DNS或WINS服务器配置有问题， 可以在HOSTS文件（HOSTS 文件在system32\drivers\etc目录下）中手工加入IP地址和服务器对如下：

169.254.173.244  MySQLserver

如果连ping IP 地址都有问题， 那么得好好检查网络的配置包括硬件的连接。 在服务器上和客户端都使用 “ipconfig/all”命令检查服务器和客户端是否在同样的网络上。

**步骤4：使用TELNET命令检查SQL 监听的端口。**

要验证SQL Server 监听的端口， 可以使用TELNET命令。假设SQL Server的IP地址是192.168.1.1，端口是1234， 那么可以运行如下命令：

TELNET 192.168.1.1 1234

如果TELNET成功，那么结果将是一个只有光标在闪的黑色屏幕。如果不成功， 那么你会得到出错的信息。需要根据这些出错信息继续排查问题。

**步骤 5：检查登录用户的SQL Server访问权限。**

和命名管道一样，需要确保客户端登录（login）帐号有权限访问SQL Server。有关这方面内容请参考命名管道篇。

需要注意的是，如果你使用Windows 系统帐号而不是SQL Server 本身的login， 那么和命名管道一样，你需要有访问服务器资源的权限。如果该Windows系统帐号不能通过Windows的认证，自然不能访问SQL server. 如果怀疑是Windows系统帐号的权限问题，可以使用SQL Server 的login 如sa尝试登录。