**一、启用TCP端口**

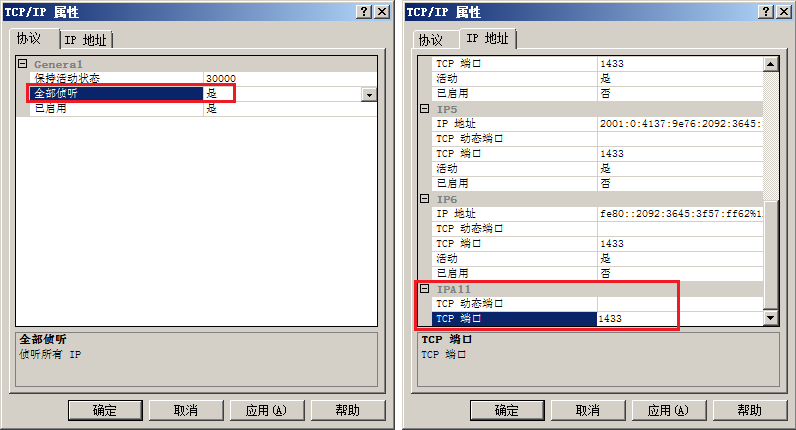
**1、启用TCP/IP协议**

　　打开SSCM（SQL Server Configuration Manager）。默认情况下，开发版、评估版、Express版禁用了TCP/IP协议。

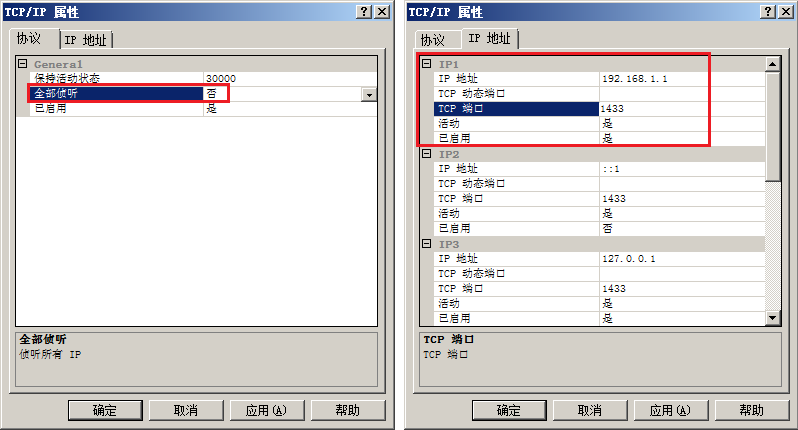
[](http://img1.51cto.com/attachment/201301/092720644.png)

**2、全部侦听**

　　这个选项用来指示 SQL Server 是否侦听所有绑定到计算机网卡的 IP 地址。如果设置为“是”，则 IPAll 属性框的设置将应用于所有 IP 地址（无论IPALL指定的是静态端口还是动态端口，动态端口每次重启的时候会改变-有缓存，重启时优先使用上次使用过的端口，如果该端口已经被占用的话则会选择其他的空闲端口）。默认值为“是”。

[](http://img1.51cto.com/attachment/201301/093444942.png)

　　如果设置为“否”，则使用每个 IP 地址各自的属性对话框对各个 IP 地址进行配置（此时IPALL不会起作用）。

[](http://img1.51cto.com/attachment/201301/093810669.png)

**3、SQL Server Browser**

[](http://img1.51cto.com/attachment/201301/094205215.png)

　　在SQL Server 2000 以前的版本中，在一台服务器上只能有一个SQL Server 安装，相当于只能安装一个默认的实例。此时，SQL Server 始终只侦听TCP1433 端口或命名管道\sql\query。

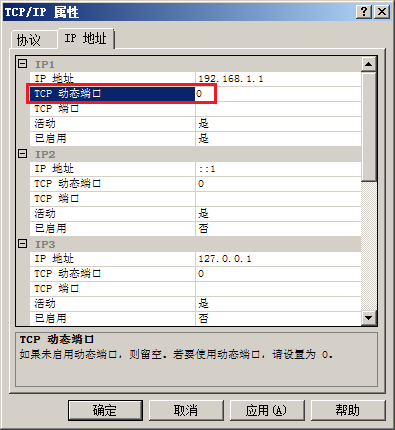
　　从SQL Server 2000 开始，引入了“实例”的概念，在一台服务器上可以有多个SQL Server 安装。而TCP1433 或命名管道\sql\query 只能被一个连接使用，一般分配给默认实例。为了解决端口冲突，SQL Server 2000 引入了SSRP 协议（SQL Server Resolution Protocol，即SQL Server解析协议），使用UDP1434 端口进行侦听。该侦听器用已安装的实例的名称以及实例使用的端口或命名管道来响应客户端请求。

　　从SQL Server 2005 开始，使用SQL Server Browser 代替SSRP。如果服务器上没有运行该服务，则不能向客户端提供端口号或管道，而且SSMS（或SQL Server 2000的企业管理器、查询分析器）中的枚举服务不能正常工作。

　　官网资料：  <http://msdn.microsoft.com/zh-cn/library/ms181087(v=sql.105).aspx>

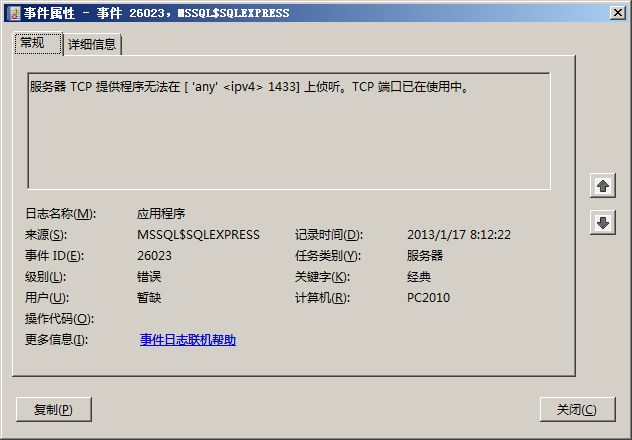
**4、动态端口**

　　命名实例在安装完成之后，默认使用动态端口。每次启动这个命名实例，都可能使用不同的TCP端口（优先使用上次运行时的端口）。可以在该命名实例启动完成之后，再进入这个页面，查看当前正在使用的端口。

[](http://img1.51cto.com/attachment/201301/095912229.png)

**5、端口冲突**

　　当SQL Server实例启动时遇到端口冲突，将导致实例的启动失败。可在系统中查到以下事件。

[](http://img1.51cto.com/attachment/201301/100418602.png)

[](http://img1.51cto.com/attachment/201301/100425871.png)

[](http://img1.51cto.com/attachment/201301/100432912.png)

[](http://img1.51cto.com/attachment/201301/100440454.png)

[](http://img1.51cto.com/attachment/201301/100448284.png)

**二、访问远程默认实例**

　　本实验从客户端192.168.1.2去访问SQL Server服务器192.168.1.1的默认实例。同时在客户端使用Microsoft 推荐的Network Monitor 3.4进行网络抓包分析。

**1、登录到SQL Server 默认实例**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ​ | Source | Destination | 协议 | 描述 | Conv ID |
| 1 | 192.168.1.2 | 192.168.1.1 | TCP | TCP:Flags=......S., SrcPort=49157, DstPort=1433, PayloadLen=0, Seq=2592455720, Ack=0, Win=8192 ( Negotiating scale factor 0x8 ) = 8192 | {TCP:2, IPv4:1} |
| 2 | 192.168.1.1 | 192.168.1.2 | TCP | TCP:Flags=...A..S., SrcPort=1433, DstPort=49157, PayloadLen=0, Seq=1781952765, Ack=2592455721, Win=8192 ( Negotiated scale factor 0x8 ) = 2097152 | {TCP:2, IPv4:1} |
| 3 | 192.168.1.2 | 192.168.1.1 | TCP | TCP:Flags=...A...., SrcPort=49157, DstPort=1433, PayloadLen=0, Seq=2592455721, Ack=1781952766, Win=256 (scale factor 0x8) = 65536 | {TCP:2, IPv4:1} |
| 4 | 192.168.1.2 | 192.168.1.1 | TDS | TDS:Prelogin, Version = 7.1 (0x71000001), SPID = 0, PacketID = 0, Flags=...AP..., SrcPort=49157, DstPort=1433, PayloadLen=52, Seq=2592455721 - 2592455773, Ack=1781952766, Win=65536 | {TDS:3, TCP:2, IPv4:1} |
| 5 | 192.168.1.1 | 192.168.1.2 | TDS | TDS:Response, Version = 7.1 (0x71000001), SPID = 0, PacketID = 1, Flags=...AP..., SrcPort=1433, DstPort=49157, PayloadLen=37, Seq=1781952766 - 1781952803, Ack=2592455773, Win=65536 | {TDS:3, TCP:2, IPv4:1} |
| 6 | 192.168.1.2 | 192.168.1.1 | TLS | TLS:TLS Rec Layer-1 HandShake: Client Hello. | {TLS:5, SSLVersionSelector:4, TDS:3, TCP:2, IPv4:1} |
| 7 | 192.168.1.1 | 192.168.1.2 | TLS | TLS:TLS Rec Layer-1 HandShake: Server Hello. Certificate. Server Hello Done. | {TLS:5, SSLVersionSelector:4, TDS:3, TCP:2, IPv4:1} |
| 8 | 192.168.1.2 | 192.168.1.1 | TLS | TLS:TLS Rec Layer-1 HandShake: Client Key Exchange.; TLS Rec Layer-2 Cipher Change Spec; TLS Rec Layer-3 HandShake: Encrypted Handshake Message. | {TLS:5, SSLVersionSelector:4, TDS:3, TCP:2, IPv4:1} |
| 9 | 192.168.1.1 | 192.168.1.2 | TLS | TLS:TLS Rec Layer-1 Cipher Change Spec; TLS Rec Layer-2 HandShake: Encrypted Handshake Message. | {TLS:5, SSLVersionSelector:4, TDS:3, TCP:2, IPv4:1} |
| 10 | 192.168.1.2 | 192.168.1.1 | TLS | TLS:TLS Rec Layer-1 SSL Application Data | {TLS:5, SSLVersionSelector:4, TDS:3, TCP:2, IPv4:1} |
| 11 | 192.168.1.1 | 192.168.1.2 | TDS | TDS:Response, Version = 7.1 (0x71000001), SPID = 51, PacketID = 1, Flags=...AP..., SrcPort=1433, DstPort=49157, PayloadLen=277, Seq=1781953480 - 1781953757, Ack=2592456290, Win=65024 | {TDS:3, TCP:2, IPv4:1} |
| 12 | 192.168.1.2 | 192.168.1.1 | TCP | TCP:Flags=...A...., SrcPort=49157, DstPort=1433, PayloadLen=0, Seq=2592456290, Ack=1781953757, Win=252 (scale factor 0x8) = 64512 | {TCP:2, IPv4:1} |

**2、从SQL Server 注销**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ​ | Source | Destination | 协议 | 描述 | Conv ID |
| 1 | 192.168.1.2 | 192.168.1.1 | TCP | TCP:Flags=...A...F, SrcPort=49157, DstPort=1433, PayloadLen=0, Seq=2592456290, Ack=1781953757, Win=252 (scale factor 0x8) = 64512 | {TCP:2, IPv4:1} |
| 2 | 192.168.1.1 | 192.168.1.2 | TCP | TCP:Flags=...A...., SrcPort=1433, DstPort=49157, PayloadLen=0, Seq=1781953757, Ack=2592456291, Win=254 (scale factor 0x8) = 65024 | {TCP:2, IPv4:1} |
| 3 | 192.168.1.1 | 192.168.1.2 | TCP | TCP:Flags=...A...F, SrcPort=1433, DstPort=49157, PayloadLen=0, Seq=1781953757, Ack=2592456291, Win=254 (scale factor 0x8) = 65024 | {TCP:2, IPv4:1} |
| 4 | 192.168.1.2 | 192.168.1.1 | TCP | TCP:Flags=...A...., SrcPort=49157, DstPort=1433, PayloadLen=0, Seq=2592456291, Ack=1781953758, Win=252 (scale factor 0x8) = 64512 | {TCP:2, IPv4:1} |

**三、访问远程命名实例**

　　本实验从客户端192.168.1.2去访问SQL Server服务器192.168.1.1的命名实例。

**1、关闭UDP1434端口或停止SQL Server Browser导致无法连接**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ​ | Source | Destination | 协议 | 描述 | Conv ID |
| 1 | 192.168.1.2 | 192.168.1.1 | SSRP | SSRP:Windows stub parser: Requires full Common parsers. See the "How Do I Change Parser Set Options(Version 3.3 or before) or Configure Parser Profile (Version 3.4)" help topic for tips on loading these parser sets. | {UDP:1, IPv4:1} |
| 2 | 192.168.1.2 | 192.168.1.1 | TCP | TCP:Flags=......S., SrcPort=49158, DstPort=Microsoft-DS(445), PayloadLen=0, Seq=364029616, Ack=0, Win=8192 ( Negotiating scale factor 0x8 ) = 8192 | {TCP:2, IPv4:1} |
| 3 | 192.168.1.2 | 192.168.1.1 | TCP | TCP:Flags=......S., SrcPort=49159, DstPort=Microsoft-DS(445), PayloadLen=0, Seq=64791044, Ack=0, Win=8192 ( Negotiating scale factor 0x8 ) = 8192 | {TCP:3, IPv4:1} |
| 4 | 192.168.1.2 | 192.168.1.1 | TCP | TCP:[SynReTransmit #6]Flags=......S., SrcPort=49158, DstPort=Microsoft-DS(445), PayloadLen=0, Seq=364029616, Ack=0, Win=8192 ( Negotiating scale factor 0x8 ) = 8192 | {TCP:2, IPv4:1} |
| 5 | 192.168.1.2 | 192.168.1.1 | TCP | TCP:[SynReTransmit #7]Flags=......S., SrcPort=49159, DstPort=Microsoft-DS(445), PayloadLen=0, Seq=64791044, Ack=0, Win=8192 ( Negotiating scale factor 0x8 ) = 8192 | {TCP:3, IPv4:1} |
| 6 | 192.168.1.2 | 192.168.1.1 | TCP | TCP:[SynReTransmit #6]Flags=......S., SrcPort=49158, DstPort=Microsoft-DS(445), PayloadLen=0, Seq=364029616, Ack=0, Win=8192 ( Negotiating scale factor 0x8 ) = 8192 | {TCP:2, IPv4:1} |
| 7 | 192.168.1.2 | 192.168.1.1 | TCP | TCP:[SynReTransmit #7]Flags=......S., SrcPort=49159, DstPort=Microsoft-DS(445), PayloadLen=0, Seq=64791044, Ack=0, Win=8192 ( Negotiating scale factor 0x8 ) = 8192 | {TCP:3, IPv4:1} |

**2、正常连接**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ​ | Source | Destination | 协议 | 描述 | Conv ID |
| 1 | 192.168.1.1 | 192.168.1.2 | SSRP | SSRP:Windows stub parser: Requires full Common parsers. See the "How Do I Change Parser Set Options(Version 3.3 or before) or Configure Parser Profile (Version 3.4)" help topic for tips on loading these parser sets. | {UDP:1, IPv4:1} |
| 2 | 192.168.1.2 | 192.168.1.1 | TCP | TCP:Flags=......S., SrcPort=49160, DstPort=1433, PayloadLen=0, Seq=2123845617, Ack=0, Win=8192 ( Negotiating scale factor 0x8 ) = 8192 | {TCP:2, IPv4:1} |
| 3 | 192.168.1.1 | 192.168.1.2 | TCP | TCP:Flags=...A..S., SrcPort=1433, DstPort=49160, PayloadLen=0, Seq=193862757, Ack=2123845618, Win=8192 ( Negotiated scale factor 0x8 ) = 2097152 | {TCP:2, IPv4:1} |
| 4 | 192.168.1.2 | 192.168.1.1 | TCP | TCP:Flags=...A...., SrcPort=49160, DstPort=1433, PayloadLen=0, Seq=2123845618, Ack=193862758, Win=256 (scale factor 0x8) = 65536 | {TCP:2, IPv4:1} |
| 5 | 192.168.1.2 | 192.168.1.1 | TDS | TDS:Prelogin, Version = 7.1 (0x71000001), SPID = 0, PacketID = 0, Flags=...AP..., SrcPort=49160, DstPort=1433, PayloadLen=51, Seq=2123845618 - 2123845669, Ack=193862758, Win=65536 | {TDS:3, TCP:2, IPv4:1} |
| 6 | 192.168.1.1 | 192.168.1.2 | TDS | TDS:Response, Version = 7.1 (0x71000001), SPID = 0, PacketID = 1, Flags=...AP..., SrcPort=1433, DstPort=49160, PayloadLen=37, Seq=193862758 - 193862795, Ack=2123845669, Win=65536 | {TDS:3, TCP:2, IPv4:1} |
| 7 | 192.168.1.2 | 192.168.1.1 | TLS | TLS:TLS Rec Layer-1 HandShake: Encrypted Handshake Message. | {TLS:5, SSLVersionSelector:4, TDS:3, TCP:2, IPv4:1} |
| 8 | 192.168.1.1 | 192.168.1.2 | TLS | TLS:TLS Rec Layer-1 HandShake: Encrypted Handshake Message. | {TLS:5, SSLVersionSelector:4, TDS:3, TCP:2, IPv4:1} |
| 9 | 192.168.1.2 | 192.168.1.1 | TLS | TLS:TLS Rec Layer-1 HandShake: Encrypted Handshake Message.; TLS Rec Layer-2 Cipher Change Spec; TLS Rec Layer-3 HandShake: Encrypted Handshake Message. | {TLS:5, SSLVersionSelector:4, TDS:3, TCP:2, IPv4:1} |
| 10 | 192.168.1.1 | 192.168.1.2 | TLS | TLS:TLS Rec Layer-1 Cipher Change Spec; TLS Rec Layer-2 HandShake: Encrypted Handshake Message. | {TLS:5, SSLVersionSelector:4, TDS:3, TCP:2, IPv4:1} |
| 11 | 192.168.1.2 | 192.168.1.1 | TLS | TLS:TLS Rec Layer-1 SSL Application Data | {TLS:5, SSLVersionSelector:4, TDS:3, TCP:2, IPv4:1} |
| 12 | 192.168.1.1 | 192.168.1.2 | TDS | TDS:Response, Version = 7.1 (0x71000001), SPID = 52, PacketID = 1, Flags=...AP..., SrcPort=1433, DstPort=49160, PayloadLen=321, Seq=193863472 - 193863793, Ack=2123846202, Win=65024 | {TDS:3, TCP:2, IPv4:1} |
| 13 | 192.168.1.2 | 192.168.1.1 | TCP | TCP:Flags=...A...., SrcPort=49160, DstPort=1433, PayloadLen=0, Seq=2123846202, Ack=193863793, Win=252 (scale factor 0x8) = 64512 | {TCP:2, IPv4:1} |
| 14 | 192.168.1.2 | 192.168.1.1 | TCP | TCP:[Keep alive]Flags=...A...., SrcPort=49160, DstPort=1433, PayloadLen=1, Seq=2123846201 - 2123846202, Ack=193863793, Win=252 (scale factor 0x8) = 64512 | {TCP:2, IPv4:1} |
| 15 | 192.168.1.1 | 192.168.1.2 | TCP | TCP:[Keep alive ack]Flags=...A...., SrcPort=1433, DstPort=49160, PayloadLen=0, Seq=193863793, Ack=2123846202, Win=254 (scale factor 0x8) = 65024 | {TCP:2, IPv4:1} |
| 16 | 192.168.1.1 | 192.168.1.2 | TCP | TCP:[Keep alive]Flags=...A...., SrcPort=1433, DstPort=49160, PayloadLen=1, Seq=193863792 - 193863793, Ack=2123846202, Win=254 (scale factor 0x8) = 65024 | {TCP:2, IPv4:1} |
| 17 | 192.168.1.2 | 192.168.1.1 | TCP | TCP:[Keep alive ack]Flags=...A...., SrcPort=49160, DstPort=1433, PayloadLen=0, Seq=2123846202, Ack=193863793, Win=252 (scale factor 0x8) = 64512 | {TCP:2, IPv4:1} |

**3、从SQL Server 注销**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ​ | Source | Destination | 协议 | 描述 | Conv ID |
| 1 | 192.168.1.2 | 192.168.1.1 | TCP | TCP:[Keep alive]Flags=...A...., SrcPort=49160, DstPort=1433, PayloadLen=1, Seq=2123846201 - 2123846202, Ack=193863793, Win=252 (scale factor 0x8) = 64512 | {TCP:2, IPv4:1} |
| 2 | 192.168.1.1 | 192.168.1.2 | TCP | TCP:[Keep alive ack]Flags=...A...., SrcPort=1433, DstPort=49160, PayloadLen=0, Seq=193863793, Ack=2123846202, Win=254 (scale factor 0x8) = 65024 | {TCP:2, IPv4:1} |
| 3 | 192.168.1.1 | 192.168.1.2 | TCP | TCP:[Keep alive]Flags=...A...., SrcPort=1433, DstPort=49160, PayloadLen=1, Seq=193863792 - 193863793, Ack=2123846202, Win=254 (scale factor 0x8) = 65024 | {TCP:2, IPv4:1} |
| 4 | 192.168.1.2 | 192.168.1.1 | TCP | TCP:[Keep alive ack]Flags=...A...., SrcPort=49160, DstPort=1433, PayloadLen=0, Seq=2123846202, Ack=193863793, Win=252 (scale factor 0x8) = 64512 | {TCP:2, IPv4:1} |
| 5 | 192.168.1.2 | 192.168.1.1 | TCP | TCP:[Keep alive ack]Flags=...A...F, SrcPort=49160, DstPort=1433, PayloadLen=0, Seq=2123846202, Ack=193863793, Win=252 (scale factor 0x8) = 64512 | {TCP:2, IPv4:1} |
| 6 | 192.168.1.1 | 192.168.1.2 | TCP | TCP:Flags=...A...., SrcPort=1433, DstPort=49160, PayloadLen=0, Seq=193863793, Ack=2123846203, Win=254 (scale factor 0x8) = 65024 | {TCP:2, IPv4:1} |
| 7 | 192.168.1.1 | 192.168.1.2 | TCP | TCP:Flags=...A...F, SrcPort=1433, DstPort=49160, PayloadLen=0, Seq=193863793, Ack=2123846203, Win=254 (scale factor 0x8) = 65024 | {TCP:2, IPv4:1} |
| 8 | 192.168.1.2 | 192.168.1.1 | TCP | TCP:Flags=...A...., SrcPort=49160, DstPort=1433, PayloadLen=0, Seq=2123846203, Ack=193863794, Win=252 (scale factor 0x8) = 64512 | {TCP:2, IPv4:1} |

**四、实验：客户端访问**

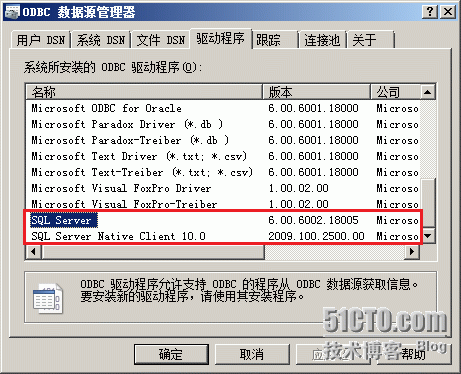
**1. 实验环境-多实例**

　　localhost安装了默认实例MSSQLSERVER，端口1433；还安装了一个命名实例SQLEXPRESS，启动本实例后发现实际使用端口1031。

**2. 连接字符串**

　　本实验使用的驱动程序为2种：OLE DB Provider for SQL Server 是Windows 操作系统集成的驱动程序，俗称ADO for SQL Server；SQL Server Native Client 10.0 需要事先安装SQL Server 客户端。

　　可以在本机的 ODBC 设置界面查看已经安装的驱动程序。

[](http://img1.51cto.com/attachment/201308/081057581.png)

　　为避开.Net Framework的干扰，本实验使用最古老的开发工具Borland Delphi 7。添加一个 TADOConnection 控件，其 ConnectionString 字符串为以下示例之一：

（1）驱动程序使用OLE DB Provider for SQL Server。采用Windows集成身份验证。

// Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=master;Data Source=localhost

（2）驱动程序使用OLE DB Provider for SQL Server，采用SQL Server混合身份验证。

// Provider=SQLOLEDB.1;Persist Security Info=False;User ID=sa;Initial Catalog=master;Data Source=localhost

（3）驱动程序使用SQL Server Native Client 10.0。采用Windows集成身份验证。

// Provider=SQLNCLI10.1;Integrated Security=SSPI;Persist Security Info=False;User ID="";Initial Catalog=master;Data Source=localhost;Initial File Name="";Server SPN=""

（3）驱动程序使用SQL Server Native Client 10.0。采用SQL Server混合身份验证。

// Provider=SQLNCLI10.1;Integrated Security="";Persist Security Info=False;User ID=sa;Initial Catalog=master;Data Source=localhost;Initial File Name="";Server SPN=""

**3. 测试结果**

（1） 连接字符串使用“Data Source=localhost”，此时，优先使用1433端口，结果连接到localhost\MSSQLSERVER。此时不需要SQL Server Browser服务(因为相当于指定了端口)。

（2） 连接字符串使用“Data Source=localhost\sqlexpress,1433”，此时，优先使用1433端口，忽略了实例名称，结果仍然连接到localhost\MSSQLSERVER。此时不需要SQL Server Browser服务。

（3） 连接字符串使用“Data Source=localhost\sqlexpress”，此时，优先使用动态侦听，由SQL Server Browser提供端口号，结果连接到localhost\sqlexpress。此时必须启用SQL Server Browser服务。

　　关闭SQL Server Browser服务，然后继续使用连接字符串“Data Source=localhost\sqlexpress”，此时，将首先尝试使用上一次SQL Server Browser提供的端口号，因此仍然可以连接到localhost\sqlexpress。（难道是本机有缓存？）

（4） 保持停止SQL Server Browser服务。重启 sqlexpress 实例，然后查看TCP/IP属性，发现本次使用1035端口。

　　连接字符串使用“Data Source=localhost,1035”，此时，优先使用1035端口，直接连接到localhost\sqlexpress。

**4. 结论**

　　如果指定了端口，就不需要SQL Server Browser服务。

　　如果同时指定了端口和实例名，则会忽略实例名称，直接连接到指定的端口所对应的实例。