目录

目录

一起学 DNS 系列（一）基础、概览

本系列主要讲述有关 Windows 下 DNS 服务器的相关知识点，由于 DNS 和 AD 结合异常紧密，所以与 AD 相关的 DNS 应用和知识将在 AD 专栏中讲述。

此为本系列的第一节。

我想没有什么人在访问的时候会直接输入 IP 地址，取而代之的是输入一串简单，容易记忆的字符，有数字的，如 www.163.com；也有纯字母的，如 www.qq.com ，只要网络连接没问题，就可以直接访问对应的网站。但在理论上访问网址依然需要用到 IP 地址的，只是字符转换为 IP 地址这部分工作由 DNS 服务器代劳了而已，而这个转换过程对于客户端来讲是完全透明的。

在讲述 DNS 之前，先让我们来认识一下几个与域名相关的概念。

什么是域名？

域名，即 DomainName，准确是说它是由英文字母、阿拉伯数字以及横"－"组成的一串字符，且英文不区分大小写，通常一个域名可以分为主体和后缀 2 部分，各部分由一个小点.隔开。以常见的 163.com 为例，163 是这个域名的主体，而后面的.com 表示这个域名属于国际域名，常见的还有.net、.cn 等等。这两部分构成了是一个完整的域名。有朋友可能问了，那我们常见的 www.163.com 又是什么呢。从整体上来说，这是一个 URL 地址，而非域名。细化来讲，WWW 是一个主机名称，在这台主机上运行着网页服务器，当客户端输入 www.163.com 时，系统会在某个系统上查询这个地址所对应的 IP 地址，如果一切正常则会很快返回这台主机的 IP 地址，之后浏览器会向这个地址发起 HTTP 请求进行网页解析。在此过程中提到的“某个系统”就是我们要讲述的 DNS 系统。

什么是 DNS？

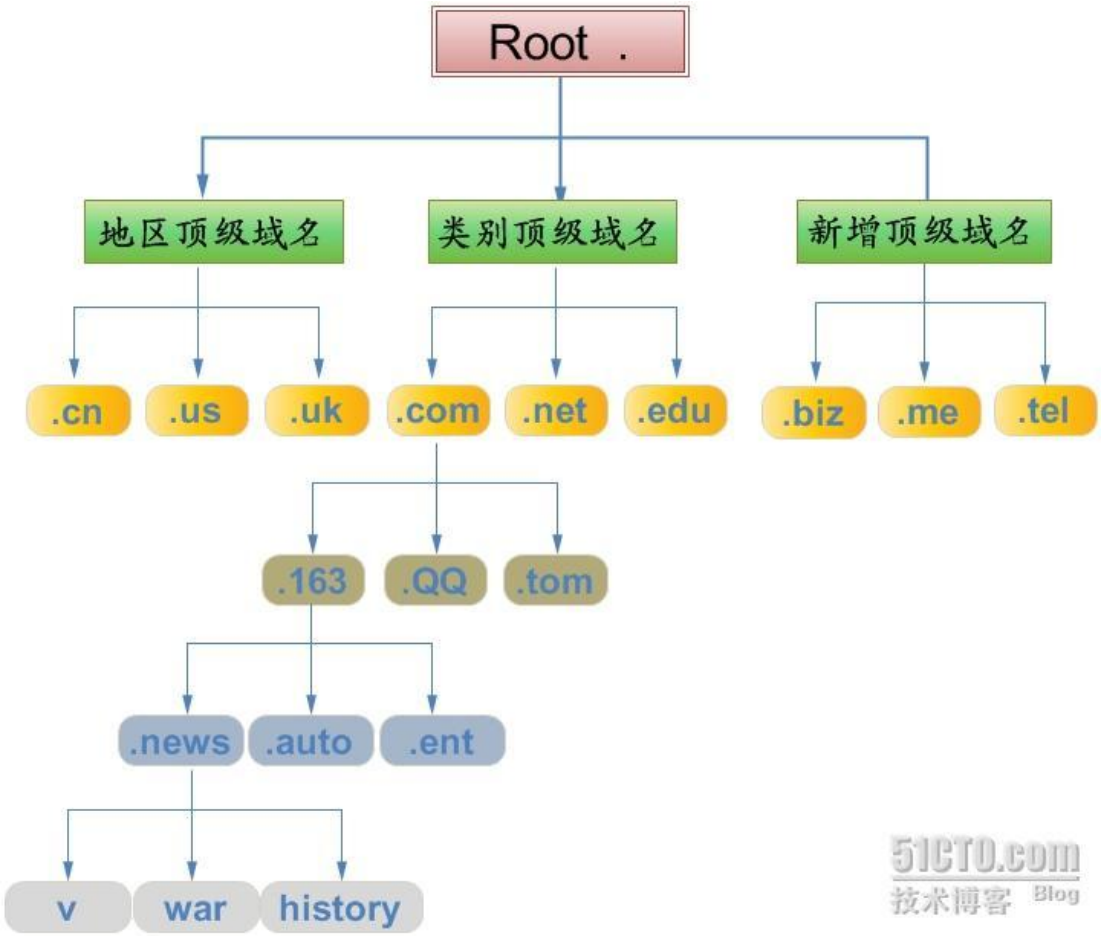
通常，我们定位一台公网计算机主要依靠的是 IP 地址，倘若这台计算机正在对外提供网页浏览服务，IP 为 1.1.1.1。访问时可以直接输入 IP 地址即可，因为这个地址简单好记，

但网络上的计算机千千万，IP 地址又是一串无序的数字，单凭人脑去记忆这些数字，显然不现实，所以急需一种途径或者方法将我们从数字苦海中解救出来，经过不断地探索，人们终于找到一个很好的解决办法，那就是通过将一些字符进行有的规律组合和拼凑，使其可以表达出一定的含义，然后将这些名词与服务器的 IP 地址对应起来，我们就可以把这些组合成为域名。当然域名的定义规范远远要比这里谈到的严谨和规范。那这部分工作或是功能由谁来实现呢，这就引出了我们本系列的主角：DNS。在今天看来，DNS 系统工作的正常与否，直接关系到整个互联网系统的安全和稳定。

DNS 即域名解析系统。前面已经提到，它可以将域名转换成 IP 地址，反之亦可。当然这也是 DNS 最基本的功能之一。

谈到 DNS 就不能不说起域名，概念可以参考上面一小节，那这两者是怎样结合在一起的呢？我们先来看一下 INTERNET 域名的整体结构。

在整个 INTERNET 网络中，域名占据着极其重要的地位，因为它有着一个很严谨的金字塔似的层级结构模式，通常又被称为命名空间。我们在阅读 DNS 相关资料时，会经常碰到“命名空间”这个词。那如何去理解呢。来看下面一张图，借此来帮助大家更好的理解命名空间的概念。



图片看不清楚？请点击这里查看原图（大图）。

从这个图中，我们可以大致了解域名体系的逻辑结构。在最顶层的是根域名，图中用一个. 表示，在此基础上延伸出其他所有的域名，严格意义上来说，我们在书写域名的时候都应该在最后带上一个.，这样才是最完整的域名表达方式，但这样明显过于繁琐，况且按照书写反而无法正常访问网页。

比如访问网易我输入 www.163.com，来看一下访问结果：

可以看到结果是“拒绝访问”，有的地址加上一个.再访问会自动跳转。

所以这里只是想让大家对此有更多了解。 在根域名下就是顶级域名，大致分为三大类，图中每一类都举出了 3 例子作为代表，其实还也很多。顶级域名位于根域名之下，而在顶级域名下又有二级域名，我们以.com 为例，请大家留意，这里说的是.com，而不是 com，单独的 com 是没有意义的，这里我省略了最后的一个点。在.com 下我列出了三个常见的二级名称，比如.163、.QQ 等，以此类推，后面可能会有很多层这样的结构。但在名称上并非是三级、四级等，而统一称为子域。下一层为上一层的子域，在.163.com 这个域名层级关系中，news 就是这与域名的子域名，同理 v 就是.news.163.com 的子域，以此类推。但我们在书写的时候是从最低一级的写起，直至顶级域名，而 DNS 查询和解析顺序正好和这个相反。

从图中我们可以看到，整个域名体系结构是非常有层次关系和等级限定的，每一个域名都有类似这样的结构，那么广泛的看，整个域名体系就是一个连续的、层级关系的逻辑的立体名称空间。知道这个概念对我们理解域名以及后面的 DNS 都非常有帮助。大家应该有这

样的反映，域名是以特定名称为起点的、立体的、名字空间。是一个很连续的结果，在这里我不厌其烦的把这些关键字眼重复再重复，目的就是为了加深大家的印象。

因为只有建立好了宏观模型，这样才可以更好的把握细节知识点。

感谢大家的支持，下一节会讲到 DNS 的安装。

一起学 DNS 系列（二）理解计算机的主 DNS 后缀选项

原本在这一节要讲解 DNS 的安装，但我发现其实在此之前，还需要向大家说明几个基本的概念，首先说到的是 DNS 后缀。

在之前的基础章节里，我们知道了一个域名可以分为主体和后缀 2 部分，这里的主机我们可以理解为一台主机或者一个网络终端的名称，后缀则直接决定这个域名的性质、类别等一些重要特征。我们这里会从单机入手，理解一下单击里的 DNS 后缀到底是怎么一回事，把这个弄清楚了，再去理解与 AD 相结合的 DNS 才会更容易。那单机的 DNS 后缀在那里找呢？我们一起来看一下：

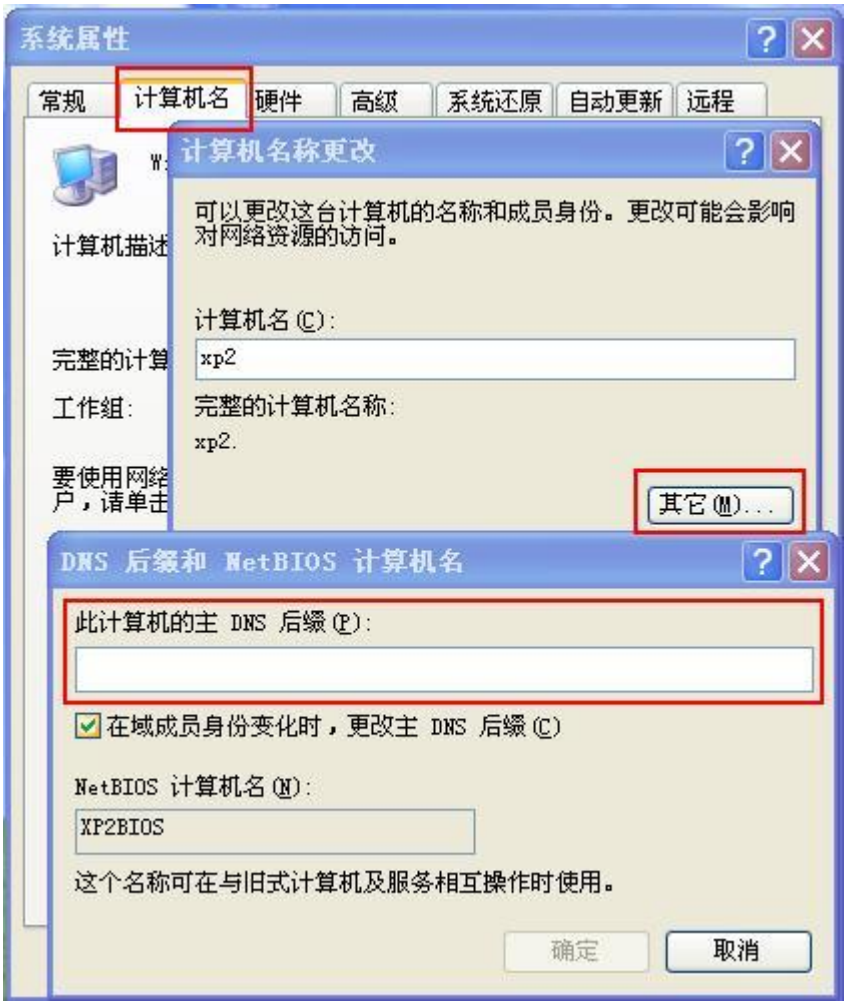
为了方便实验，我做了一个拓扑图，如下：

主机名称：win2k3
域名：os.ad
系统：Win 2003 SP2
IP地址：192.168.1.1
掩码：255.255.255.0
角色：域控制器
DHCP 服务器

主机名称：XP2
系统：Win XP SP3
IP地址：192.168.1.2
掩码：255.255.255.0
角色：单机

上图已经将两者的关系表述清楚了，XP2 现在还是一台普通的客户端，为加入域。

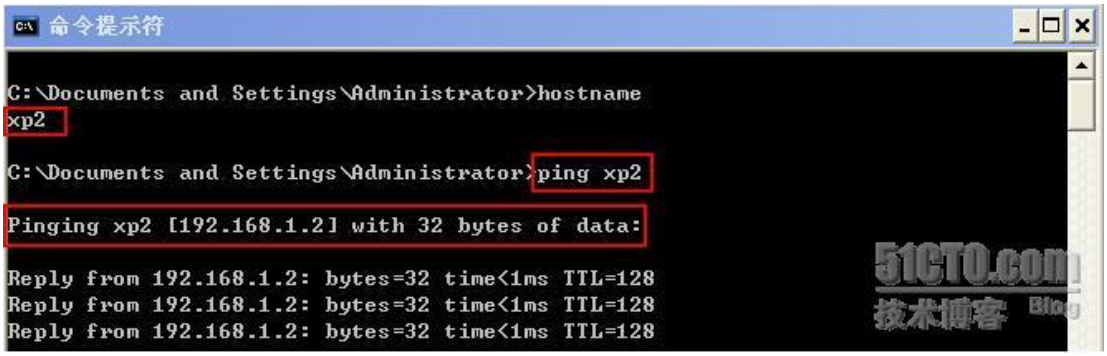
我们打开 XP2 的属性，在里面找一下它的 DNS 后缀，如下图：



从上图我们可以看到，默认情况下 计算机的 DNS 后缀的空白的。但请大家注意，图中的是主 DNS 后缀，为什么是主呢？难道 DNS 后缀还有主次之分？？其实这样的，这个里面的主 DNS 后缀是针对这台主机而言的，相比之下，每一个网卡也可以设置自己的 DNS 后缀，而那些设置准确来讲是绑定某一网卡上的连接 DNS 后缀，这部分内容在后面还会详解。

我们已经知道主 DNS 后缀的位置了，那这里的设置到底有什么作用呢？其实是这样的，我们可以利用一个命令来解答这个疑惑。PING 命令的作用我想大家都很清楚，也是作为网络连通性诊断的一个必备工具，当我们用 ping 命令去 ping 一台主机或一个地址会发生什么呢？

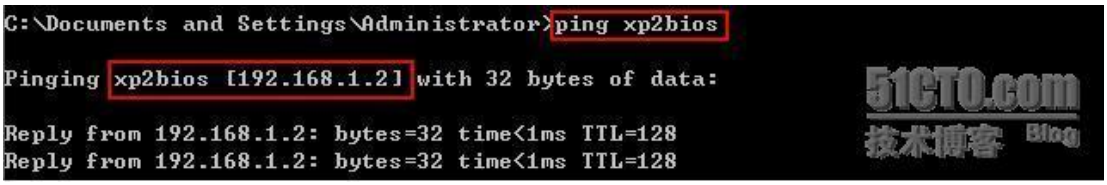
我们运行 ping xp2 命令，看一下效果：



图片看不清楚？请点击这里查看原图（大图）。

从上图得知，XP2 自动获取的 IP 是 192.168.1.2，请大家注意第三个红框，

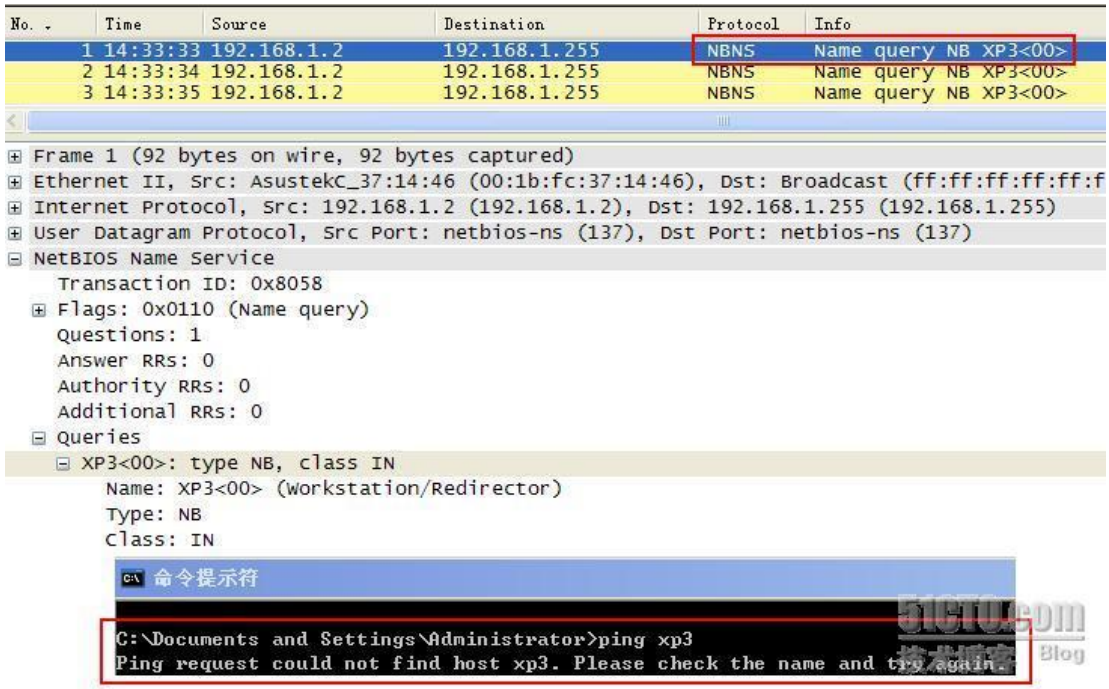
XP2　[192.168.1.2]，前者是主机名，后者是对应的 IP 地址。这个解析过程是由系统本身完成的，我们也可以 ping 一下它的 NETBIOS 名称，如下图：



图片看不清楚？请点击这里查看原图（大图）。

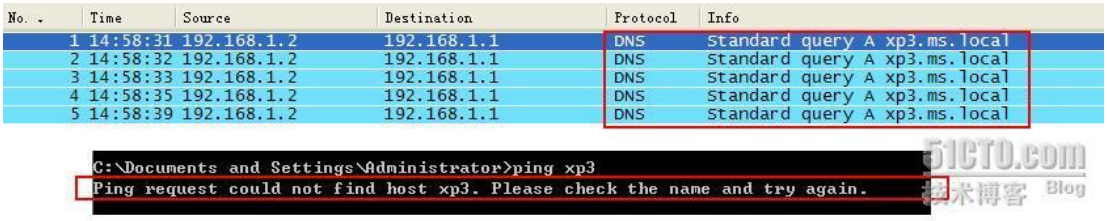
可以看到，最终结果都是一样的，但不同的显示对象有区别。这里存在着一个解析的过程，单单 ping 本机比较难分析这个解析过程，我们任意 ping 一个名称，然后用 wireshark 来监视这个过程，应该会有更多的发现。

运行 ping XP3 命令　，看一下结果：



显然，结果是不通的，通过 wireshark 的分析我们可以看到在这个过程中，系统会利用 NBNS 服务来查找是否有名为 XP3 的主机，如果没有则返回一个信息，通知查询者未能找到主机 XP3。此时，主 DNS 后缀依然为空，我们是否可以这样试想，如果主 DNS 后缀不为空，在查询时候某台主机时，是否会自动将这个后缀添加到主机名之后呢，比如 后缀为 ms.local，在查询 XP3 时，返回的结果就是 ping xp3.ms.local 呢？我们来试一下，首先将主 DNS 后缀改成 ms.local 并重启。结果如下图：

我们可以看到，计算机名也变成了 xp2.ms.local 了，这个有点类似域内的计算机名称的样式。我们现在再来 ping 一下 XP3，结果如下图：



图片看不清楚？请点击这里查看原图（大图）。

可以看到，在图形界面下仅仅提示找不到 XP3 这个主机，而从抓包的情况来分析更明显些，我们可以很清楚的看到当提交此请求后，系统会向已配置的 DNS 服务器（DHCP 自动配置）发起解析请求，来查询是否存在 XP3.ms.local 这条记录，经过多次查询后返回的结果如图，结果是未能找到 XP3 主机。从这个实例我们可以看出，在名称解析时，系统会自动加上之前已设置的主 DNS 后缀，然后参与解析。当在指定的 DNS 服务器上找不到记录时就会返回相应的信息。反之，如果没有配置 DNS 服务器地址时，系统只能通过 NBNS 服务器利用广播方式进行查找了，具体如何查找可以参考之前的 WINS 系列。

我们再结合现有的 AD 域来看，在 DC 上，肯定是无法解析到 XP3 这台主机，XP2 也一样，因为在 DC 的 DNS 上没有 XP2 和 XP3 的记录信息。如果我们将 XP2 的主 DNS 后缀改成和 AD 域一样，那么当在 XP2 上 ping win2k3 时，一定会成功，因为之前讨论过，系统会自动将其转换为 win2k3.os.ad ，因为在 192.168.1.1 这台服务器上有这条记录，所以可以 ping 同，反之在 DC 上 ping XP2 则不行。因为在 DNS 上并不存在 xp2.os.ad 的记录，之所以这个过程无法交互，是因为我们并非按照常规方法将 XP2 注册到 DNS 服务器上，而仅仅是在本机上修改了主 DNS 后缀，从某种程度上来说这仅仅是个假象而已。如果按照一般加域的步骤，相互通讯是没有问题的，这个我们后面会讨论到。

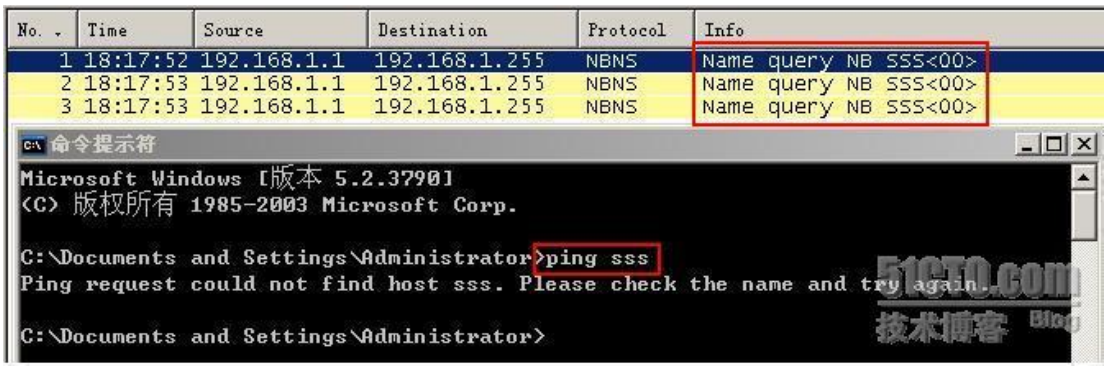
后面的章节依然是有关 DNS 后缀的内容，敬请期待！

一起学 DNS 系列（三）理解计算机的多种 DNS 后缀选项

上节介绍了计算机主 DNS 后缀的一些内容，今天这一节将继续延伸此话题，同时也谈到与之相关的其他概念。

本节的试验环境是一台 DC 和一台客户机相连，很简单的拓扑环境。如下图：

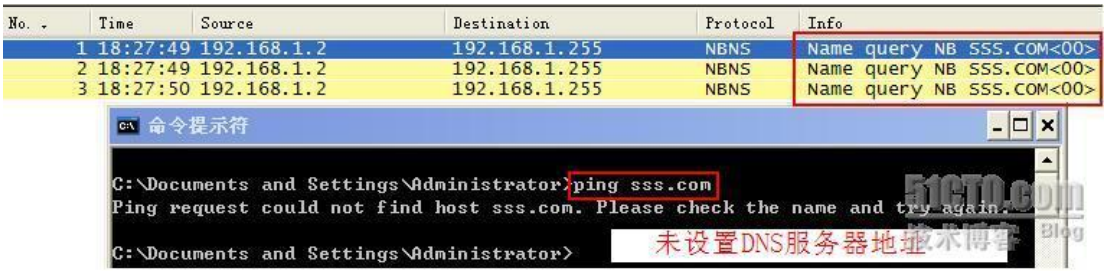
在 WINS 系列中我们说到，当利用 ping 程序访问某一名称时，如果本机设置了 WINS 服务器，则会向此服务器发出解析请求，反之，系统将会利用 NBNS 服务通过广播的方式来解析此名称。如下图：



图片看不清楚？请点击这里查看原图（大图）。

但是，当我们 ping 一个类似域名结构的名称时候，比如 ping sss.com ，此时系统会分两种情况进行处理。

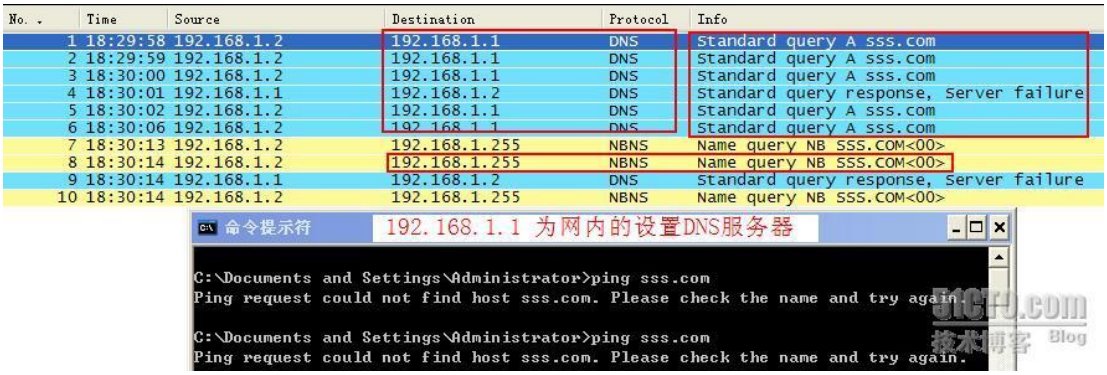
当本机网卡属性中未设置 DNS 服务器时，系统依旧会利用 NBNS 服务进行解析此名称，尽管这个名称看上去具有域名的结构特征。如下图：



图片看不清楚？请点击这里查看原图（大图）。

从上图可以看到，NBNS 服务试图在解析 sss.com 这个名词，而不是一个域名，因为对于 NBNS 服务而言，只有名称的概念，而不存在域名的概念。

当我们为其配置一个合法且有效的 DNS 服务器地址之后呢？测试结果如下图：



图片看不清楚？请点击这里查看原图（大图）。

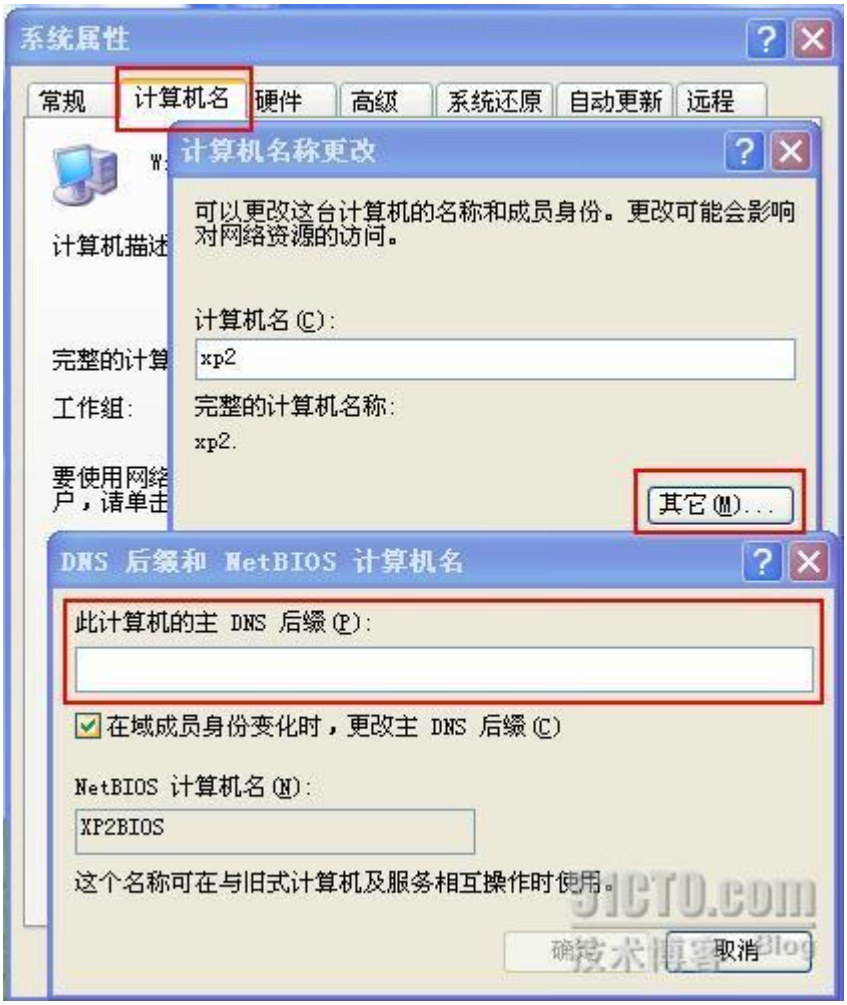
刚开始，系统会向 DNS 服务器发出标准的 A 记录查询请求，看是否存在 sss.com 这样的 A 记录，经过几次尝试以失败告终，第四个数据包是由 DNS 服务器发给客户机的，所传达的信息是“Server Failure”。可见在这台 DNS 的区域里并不存在 sss.com 这样的 A 记录。

那这些内容和我们今天要讲的多种 DNS 后缀有什么关联呢？其实是这样的，当我们为系统配置了多种的 DNS 后缀后，再去 ping 某一名称时，系统会自动在该名称后添加上已设定的 DNS 后缀，并对此名称进行解析，此时才是真正的对一个域名进行解析，其中包括了主体名称和设定的 DNS 后缀。

如何设置这些 DNS 后缀呢？在 Windows 系统中，有很多地方可以设置。下面我们一一进行描述。

设定主 DNS 后缀

从名称上应该可以看出，这个 DNS 后缀的优先级是很高的，因为这个设置是针对整个系统的，这一部分在上一节中也有提及。我们再来回顾一下设置的过程。按照下图找到如下位置：

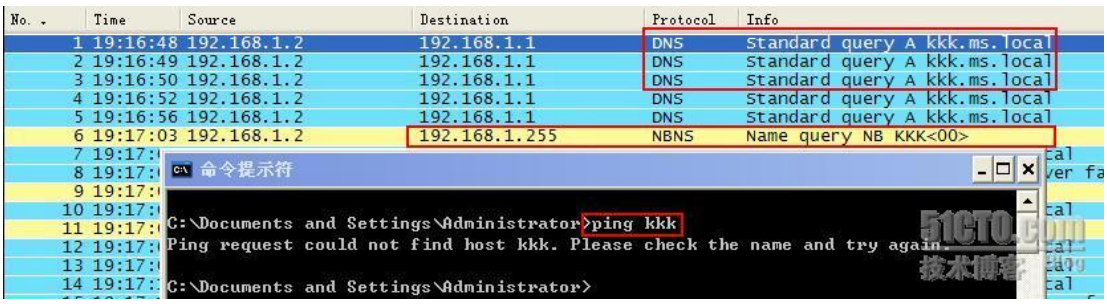


这里就是设置本机的主 DNS 后缀的地方。我们在此输入 ms.local 并重启。

下面是设置的结果：

大家可以看到，此时计算机全名已经修改成了 xp2.ms.local，其中 XP2 是原来的计算机名称。

此时，我们任意 ping 一个名称，比如 kkk，看一下与之前的有什么变化。如下图：



图片看不清楚？请点击这里查看原图（大图）。

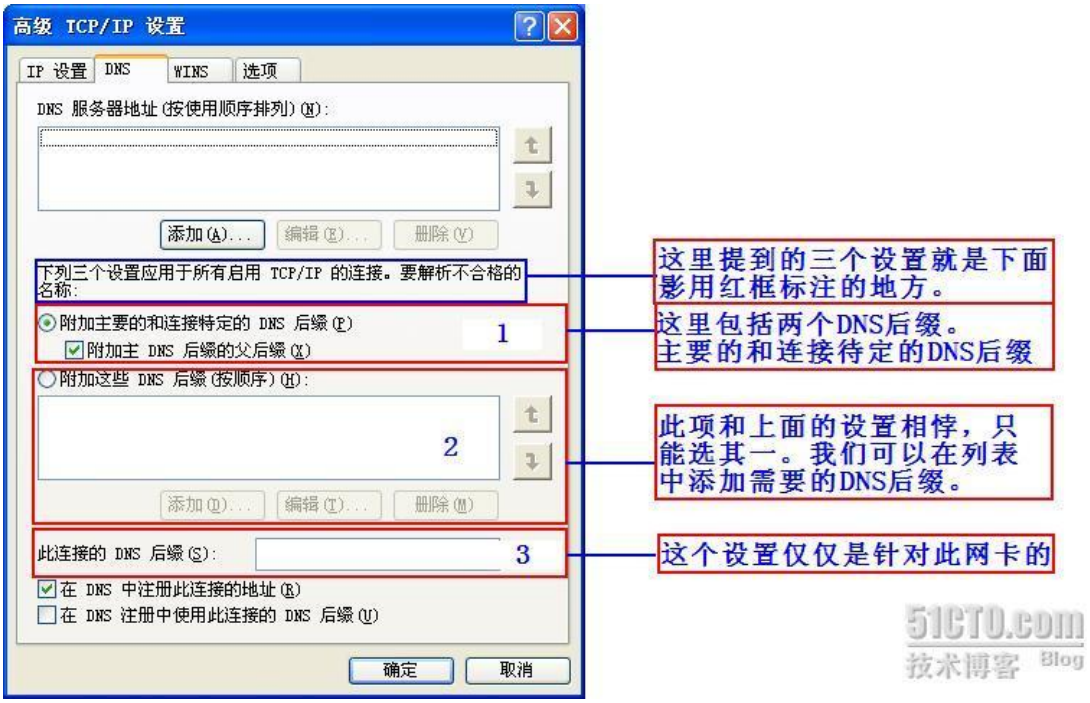
从上图可以看到，在解析 kkk 这个名称的时候会自动在后面加上 ms.local 的主 DNS 后缀进行解析，类似的，我们输入另一个名称比如 mmm，也会得到同样的结果。也就是说，

当我们输入一个单标签的名称时，系统会自动检查此时是否设置有主 DNS 后缀，如果有，则自动加上并组合成【单名称 ．主 DNS 后缀】这样的形式参与解析。

还有一点要提一下，如果我们把主 DNS 后缀改成很 DC 域名一样，那么 ping win2k3 时就可以自动替换成 ping win2k3.os.ad 了，因为 DNS 里有这条记录，所以可以顺 ping 通了。

而类似这样的单标签的名称，系统统称之为【不合格的名称】。

除了在这里设置，还有其他地方吗？ 当然有的。打开网卡属性并点选【高级】，在【DNS】选项卡中就可以看到了，如下图：



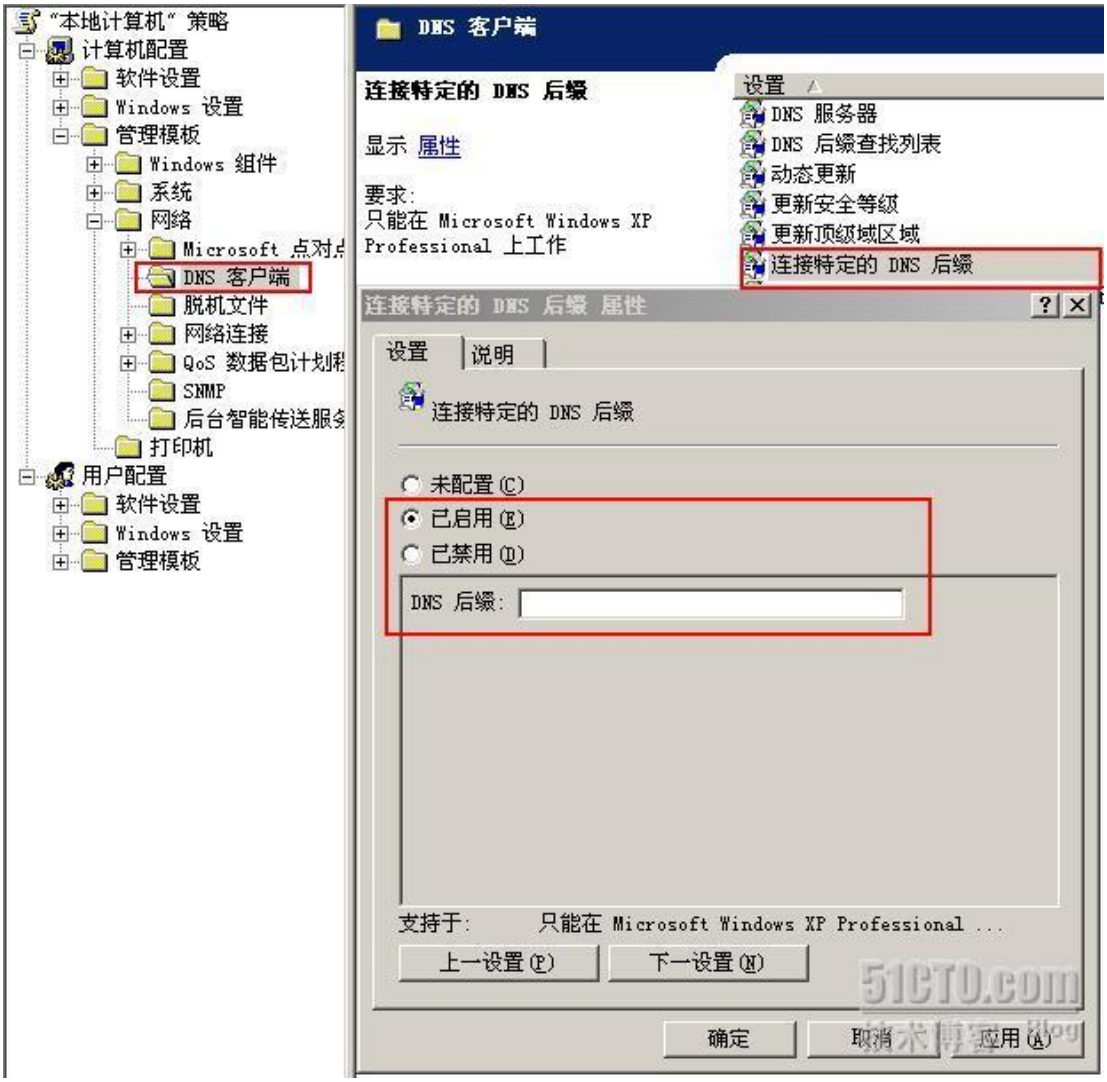
在上图中，我们可以看到系统提供了另外三个地方可以进行 DNS 后缀的设置，以此来解决单标签解析，也就是解析不合格名称的问题。

这里在解释一下，何为不合格名称。通俗的讲，当我们为系统配置了合法有效的 DNS 时，利用 ping 这个程序原本应该解析符合域名基础结构的名称，比如 sss.com，但可能在很多情况下会提交一些单标签的名称进行解析，这样会给系统带来一定的资源消耗负担。为

了解决这个问题，系统允许我们自行添加符合实际情况的 DNS 后缀，以此满足不同的 DNS 后缀解析需求。

在这个图中，有一个名称需要大家认识，那就是【连接特定的 DNS 后缀】。

当我们需要为系统制定一个有别于主 DNS 后缀的名称时，我们就可以使用一个连接待定的 DNS 后缀，这个后缀是通过系统组策略设置的，具体位置如下图示：



图片看不清楚？请点击这里查看原图（大图）。

在其中输入一个 DNS 后缀，比如 spec.com，然后确定并重启，这样才可以生效。但是前提是没有设置主 DNS 后缀，否则系统将忽略此设置。

当我们选择第二项设置时，第一部分自动变灰，此时我们需要添加一个后缀才可以生效。这里就不再演示了，操作都很简单。

第三个设置则是仅仅针对此网卡的，也就是说如果本机有多张网卡时，禁用次张网卡，则设置的 DNS 后缀也将无效，如果两张都启用，且都设置了【此链接的 DNS 后缀】，那么究竟以谁的 DNS 后缀为主呢？

当系统中同时存在两张或多张网卡时，我们可以通过【网络连接】的高级选项来设置网络服务的访问顺序。如下图示：

打开【高级设置】，如下图：

从上图中，我们可以发现通过调整 NC1 和 NC2 网卡的先后顺序来设定网络服务分配的顺序，继而确定选择使用那张网卡的 DNS 后缀设置。这一选项藏的比较隐蔽，可能很多朋友都没太注意。

实际上，我们平时一般都不怎么会用到这些设置，今天用整个篇幅来讲这个内容，目的是想扩大一下大家的视野，毕竟这些东西平时我们很少触及。

今天的内容就讲到这里，如果本文有什么遗漏或错误，请多谢补充或指正，谢谢！

一起学 DNS 系列（四）DNS 服务器、客户端安装及配置

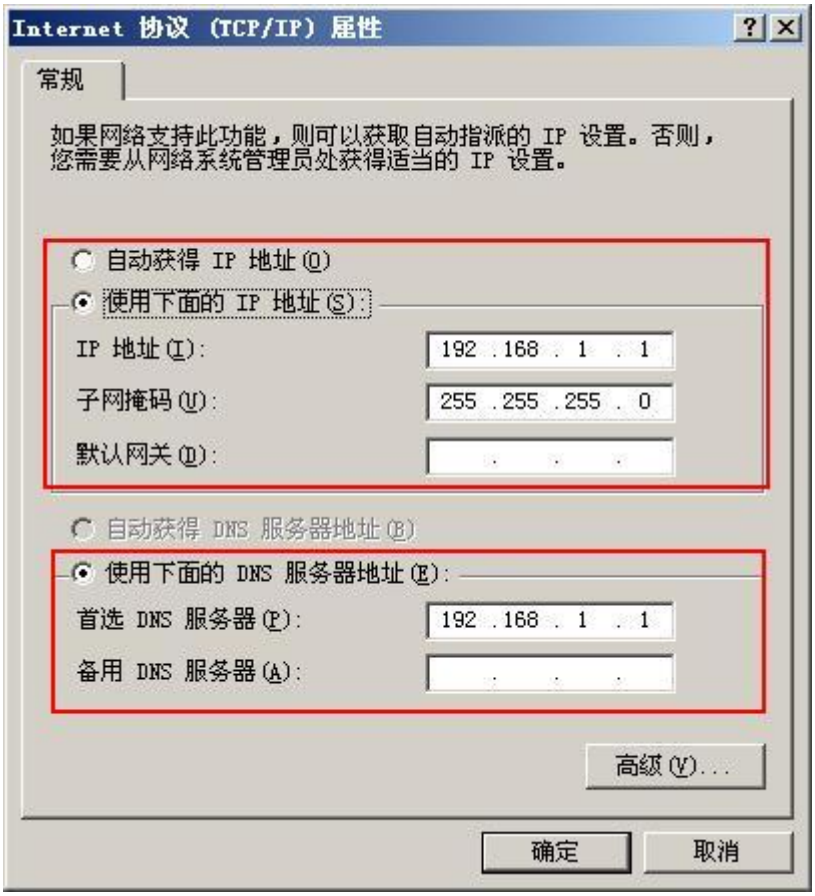
经过前三节内容的铺垫，想必大家应该对 DNS 以及相关概念有个基本的认识了，本节就开始讨论 DNS 服务器的安装和配置，同时也会有客户端的设置，内容都很基础。

今天我们的试验环境是由一台服务器，用于安装 DNS 服务，以及一台 XP 系统客户机组成。具体参数和配置如下图：

主机名称：WIN2k3
工作组：WorkGroup
系统：Win 2003 SP2
IP地址：192.168.1.1
掩码：255.255.255.0
角色：DNS、DHCP

主机名称：XP2
工作组：WorkGroup
系统：Win XP SP3
网络配置：自动获取
角色：单机

首先会在服务器安装好 DHCP 组件（具体安装过程请查阅 DHCP 系列），便于为客户端自动分配 IP、掩码以及 DNS 等参数。以下是这台台服务器的网络配置：



我们可以看到，首选的 DNS 服务器一栏内输入的是本机的 IP，因为此时本机也是 DNS 服务器。

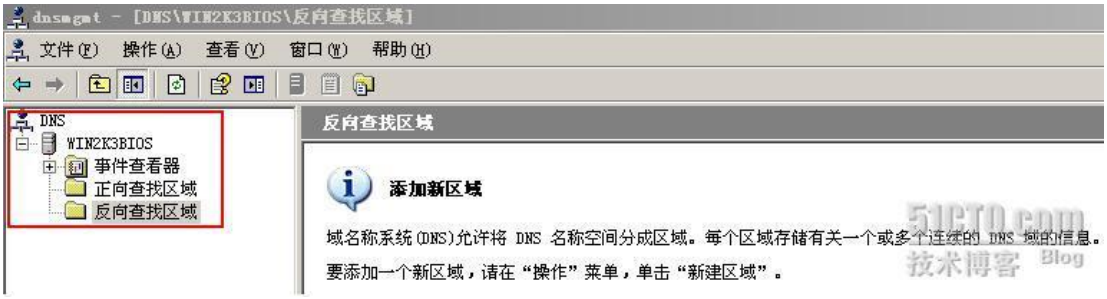
现在我们来开始安装 DNS 组件。找到控制面板/添加或删除程序/，找到如下设置：

图片看不清楚？请点击这里查看原图（大图）。

点选【域名系统】并确定后系统会自动安装，过程很简单，不需要做任何设置。

安装完成后如下图：

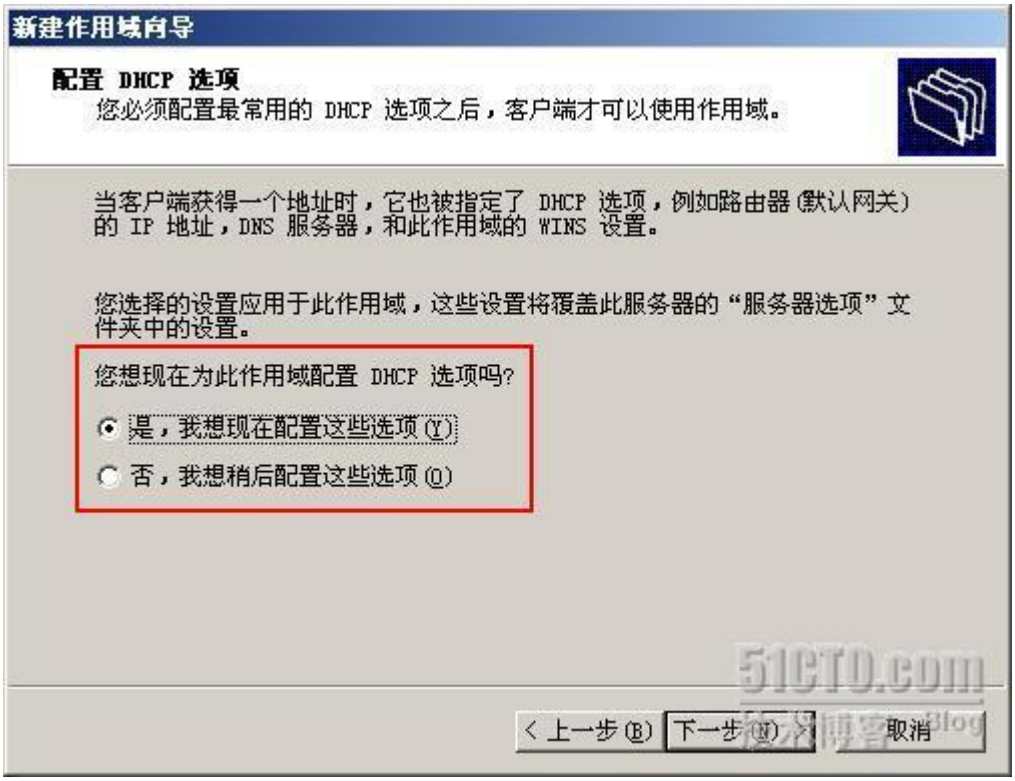
看到这个画面表示 DNS 组件已经安装成功，我们现在启动 DNS 组件。如下图：



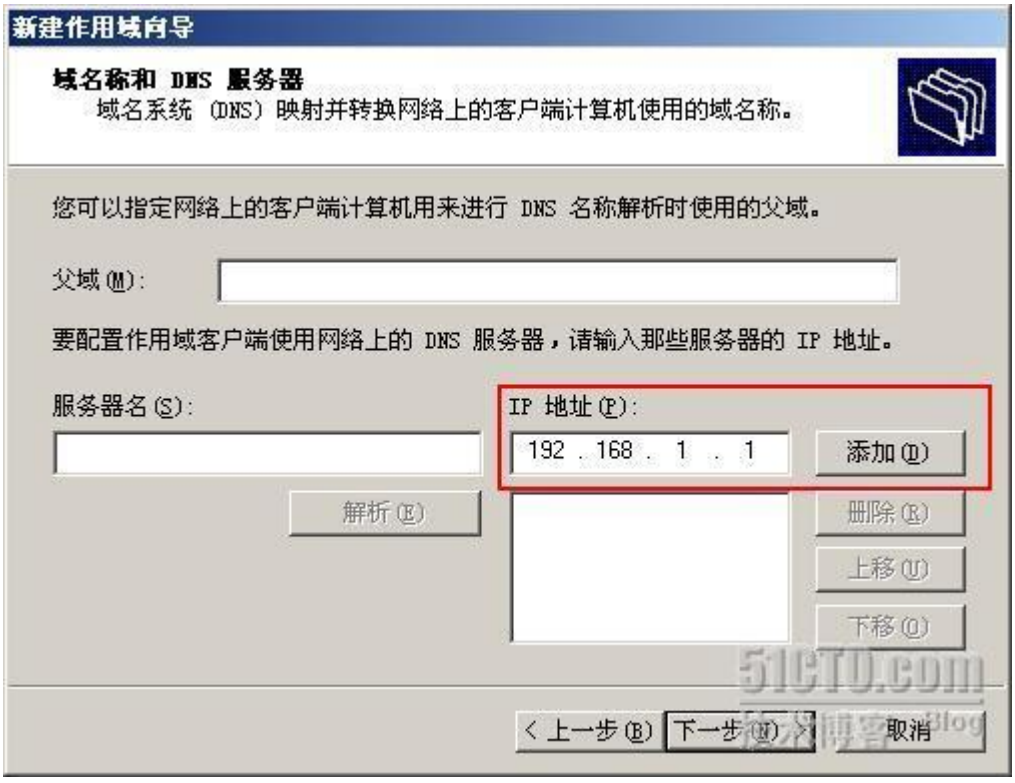
图片看不清楚？请点击这里查看原图（大图）。

这个是 DNS 组件的图形管理界面，是不是很简洁呢！哈哈，内容可不简单呢~

到此，DNS 服务器端已经安装完毕，大家可以发现过程出奇的简单。接下来，我们需要配置 DHCP，以便为客户端 XP2 自动分配 IP 和 DNS 地址。打开 DHCP 管理界面，新建一个作用域，如 192.168.1.2-192.168.1.10，因为我们需要为这个作用域配置额外选项，所以在此处需要留意。如下图：

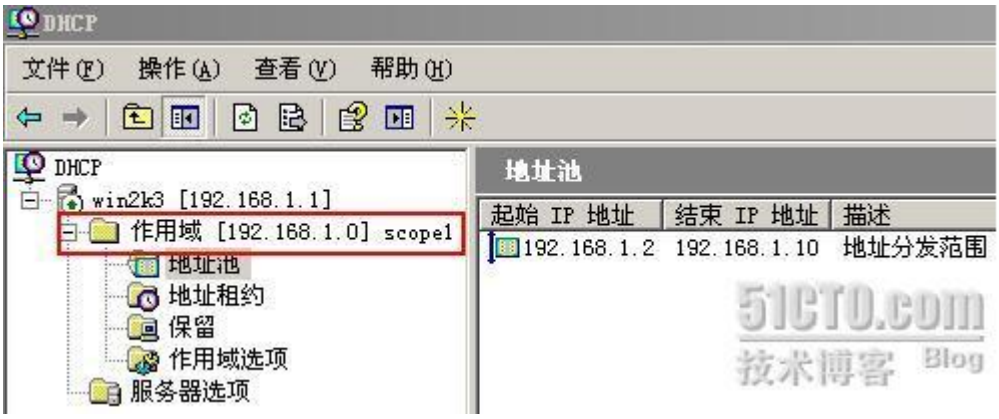


如图所示，选择第一个选项并点击【下一步】，如下图：



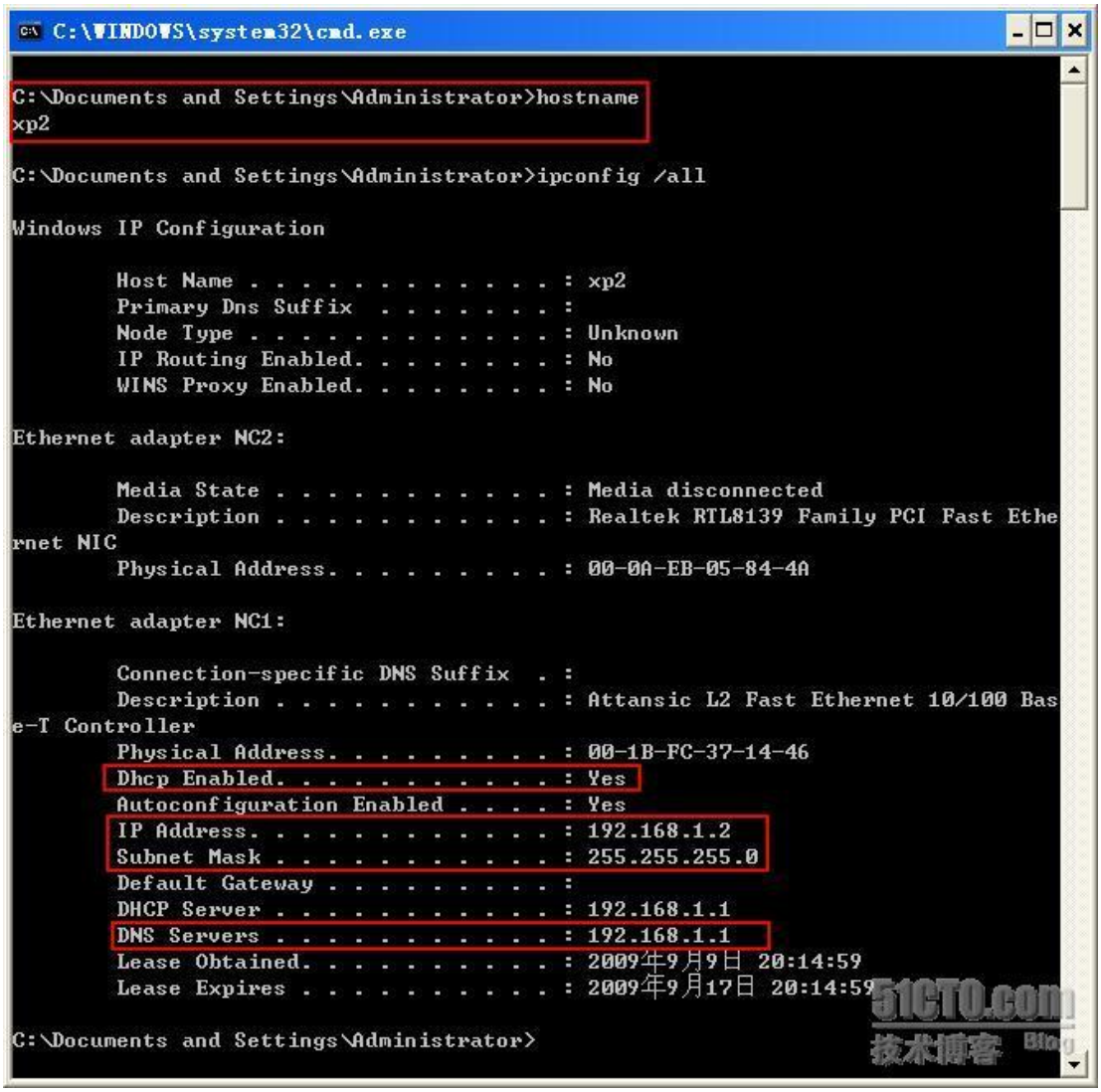
输入当前的 DNS 服务器地址并点击【添加】，最后再激活此作用域即可，如下图：

以上设置完成后，打开 DHCP 管理程序，可以看到作用域已经建立并生效。如下图：



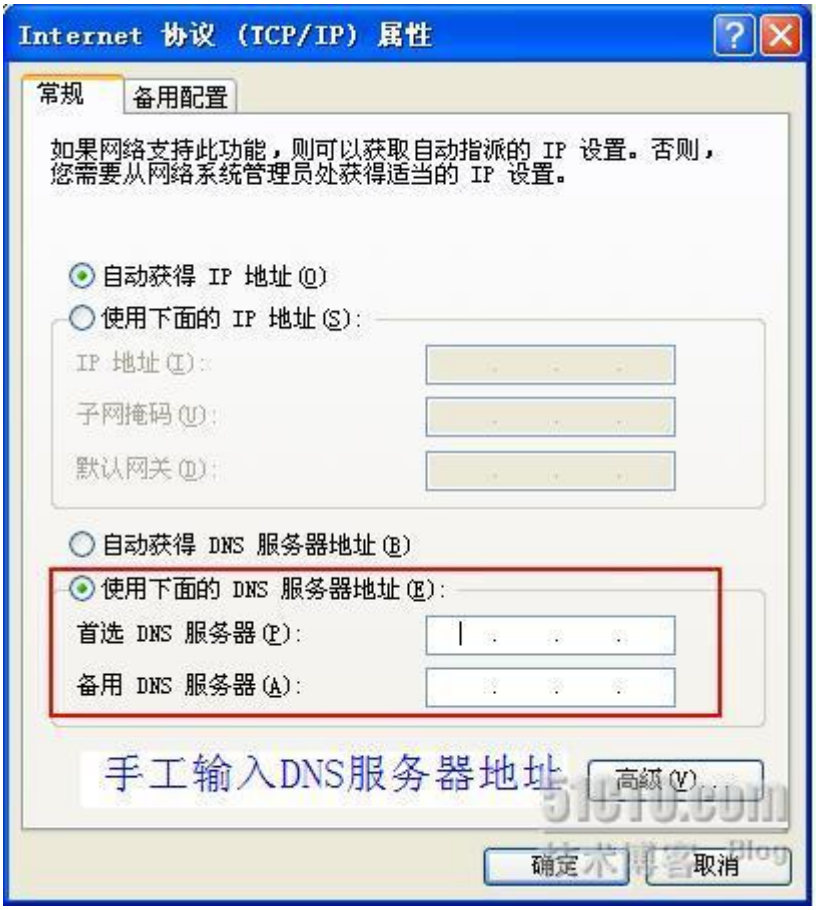
现在启动 XP2 主机，看客户端是否可以自动获取到 IP 和 DNS 服务器地址。

为了验证，我们在 XP2 上运行 CMD 命令，来查看一下当前的网络配置：



图片看不清楚？请点击这里查看原图（大图）。

可以看到，XP2 已经自动获取到了 IP 和 DNS 地址。如果内网没有 DHCP 服务器，则指需要在网卡属性里手工修改 DNS 地址即可。如下图：



纵观全篇，单纯从安装和配置的角度上来讲，其实过程都很简单。可能有的朋友问了，为什么 DNS 没有配置区域等等呢，是的，暂时还没有，因为这部分内容的知识点很多，我们会在下一节中着重讲解，敬请期待！

谢谢大家的支持！

一起学 DNS 系列（五）创建 DNS 正向、反向查找区域

上一节中简要介绍了 DNS 服务器及客户端的安装，尽管涉及的知识点很少，但也已经包含了 DNS 系统不可缺少的两部分，即 DNS 服务器和客户端，此外，DNS 系统还包括区域和资源记录，本节主要讨论 DNS 正向和反向查找区域的创建。

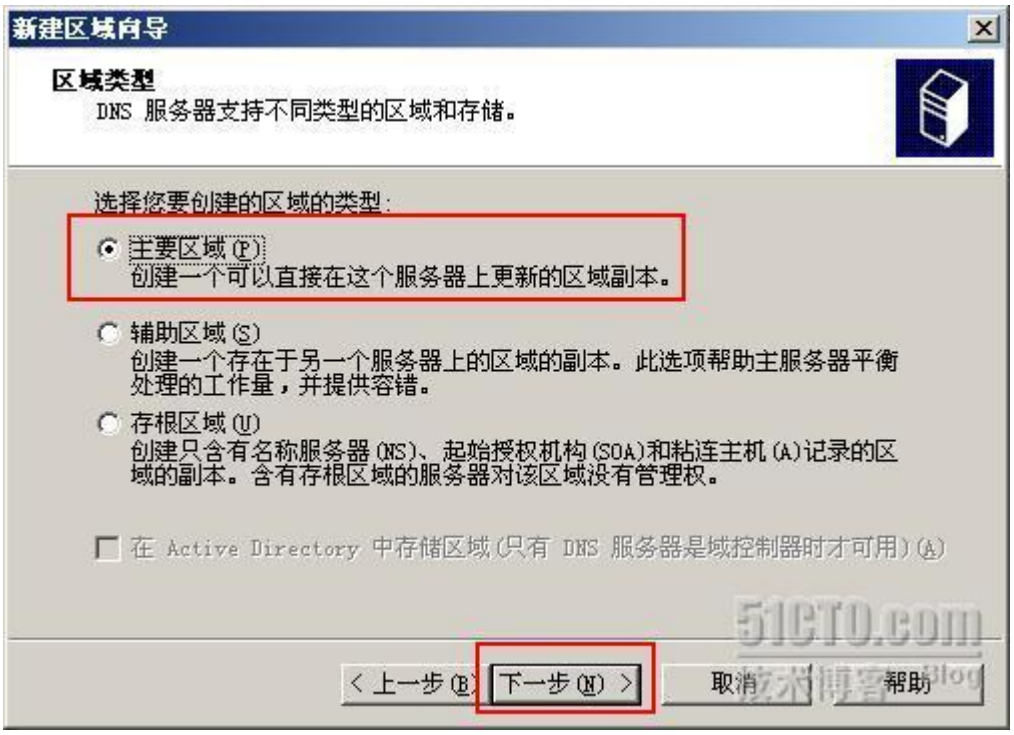
第一节里提到了“区域”这个概念，简单来讲，区域就是 DNS 服务器具有权威管理权限的连续的命名空间。在一个 DNS 服务器里，可以创建多个区域，比如 a.com 和 b.com，而在每一个区域下有可以新建多个域。请注意，这里首先建立的是区域，其后才是域，而且也只能是域，因为在一个区域里是不能再新建区域的，每一个建立的域，在 DNS 层次结构中体现出来就是一个层级，比如在区域 a.com 下，我们可以新建一个名为 b 的域，那么在此基础上新建的主机或其他记录将是类似 b.c.a.com.这样的结构。而且新建的域的名称不可以有 . ，但系统会自动将其转换为层级的域的结构。这一部分内容比较抽象，我们来做个演示。

以创建一个正向查找区域为例，所谓正向查找，也就是说在这个区域里的记录可以依据名称来查找对应的 IP 地址。右键点选【正向查找区域】，选择【新建区域】如下图：

选定后，如下图：

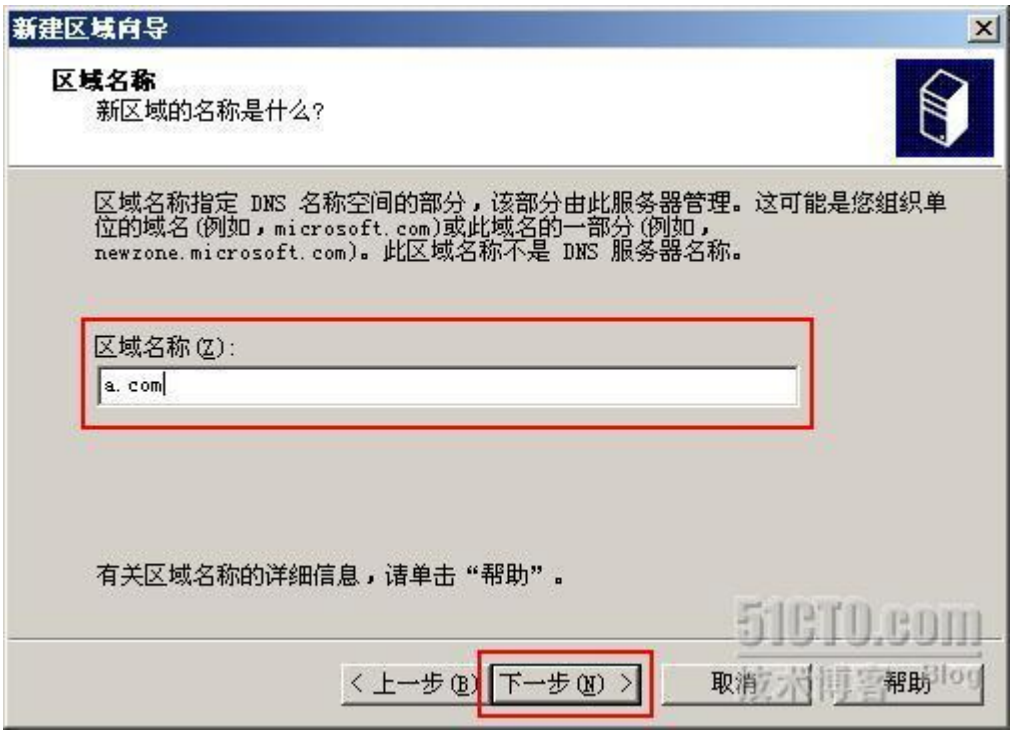
请大家留意第一个红框的内容，这里是对 DNS 区域的一个概述性的解释，也就是说，我们可以把创建出来的区域当作一个特殊的容器，里面存放着很多名称记录，而且这些名称必须符合 DNS 名称的规范和定义，否则将无法被创建。这个特殊的容器的作用在于，DNS 系统可以依据名称记录内容将名称转换成其他的数据，比如 IP 地址，或者以此为基础响应客户端的各种请求，从而提供相应的网络功能和服务。可能在大家的印象里，查找 IP 地址这类服务用的更多些，事实上也的确如此。

了解到这些后，点击【下一步】继续，如下图：



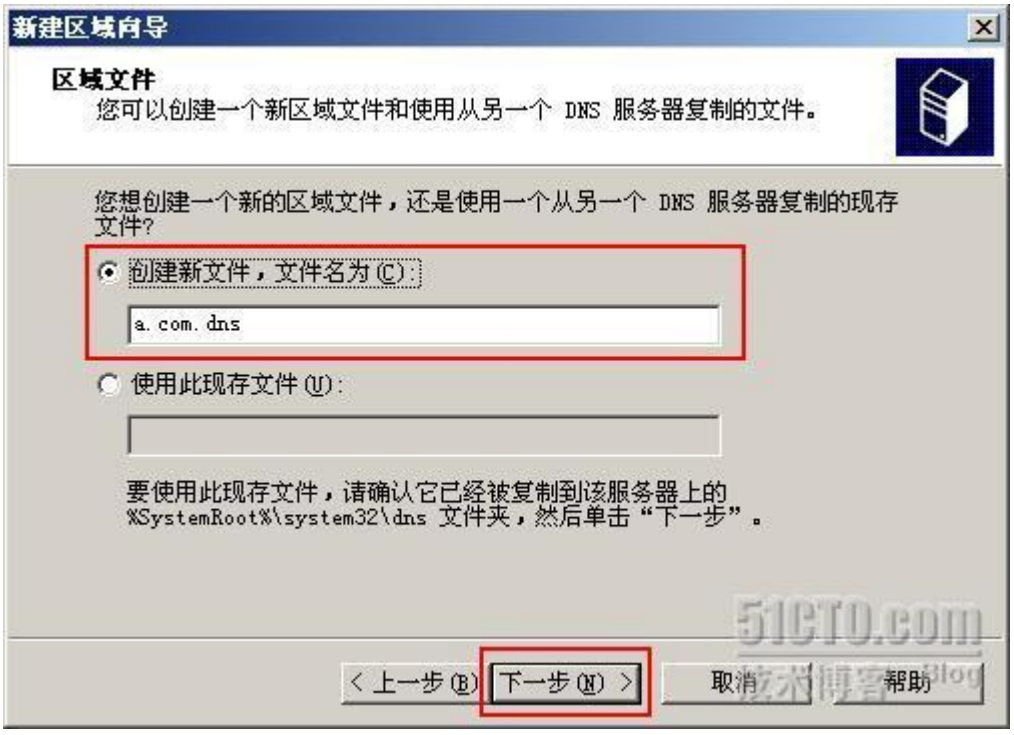
为了分散管理 DNS 区域，系统将区域划分为三种类型，分别是主要区域、辅助区域以及存根区域。在每一个区域类型下都有简要的文字说明，主要区域是包含了该命名空间内所有的资源记录，是该区域内所有域的权威 DNS 服务器。我们可以对此区域内的记录进行增删改等操作。相对的，辅助区域也可以理解为副本区域，我们可以在另一台服务器上增设辅助区域，而区域内的所有记录均来源于主要区域，辅助区域内的记录是只读的，可以响应名称解析请求，这样可以分担一部分主要区域的压力，从而起到冗余的作用。最后一个是存根区域，这个区域只含所管理区域的有 SOA、NS 以及 A Glue 记录，这部分内容会在下一节细讲。

通常，我们都会选择主要区域，然后点击【下一步】继续，如下图：



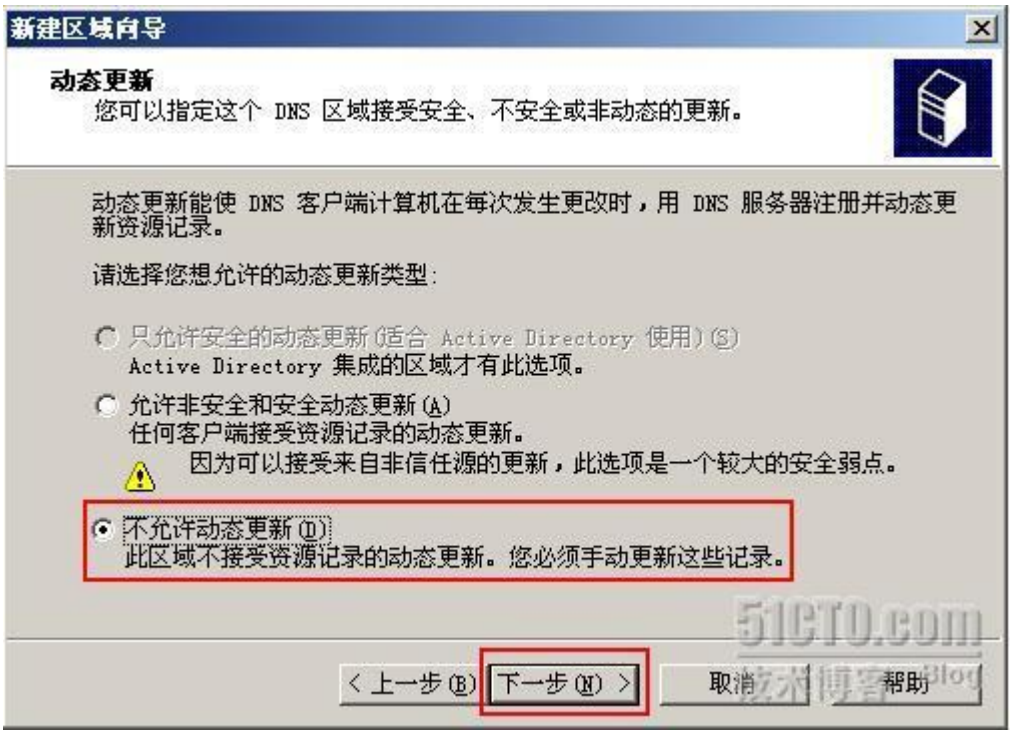
这里我们需要设定一个区域名称，请注意，为了规范起见建议大家使用标准的区域名称空间，比如这里的 a.com，同时你可以根据你的组织的域名来填写。

设定好后，点击【下一步】继续，如下图：



主要区域的记录都是以文本形式存在于本地的，具体地址可以参考最下方的说明。同时，如果已经存在这样的文件，也可以直接使用，这里我们按照默认设置。

然后点击【下一步】继续，如下图：



一般情况下，当区域并非 AD 集成区域时，默认的动态更新类型是【不允许动态更新】，点击【下一步】继续，如下图：

出现这个画面表示 a.com 区域创建完毕，向导会给出一个该区域的大致信息。安装完成后，打开 a.com 区域，如下图：

图片看不清楚？请点击这里查看原图（大图）。

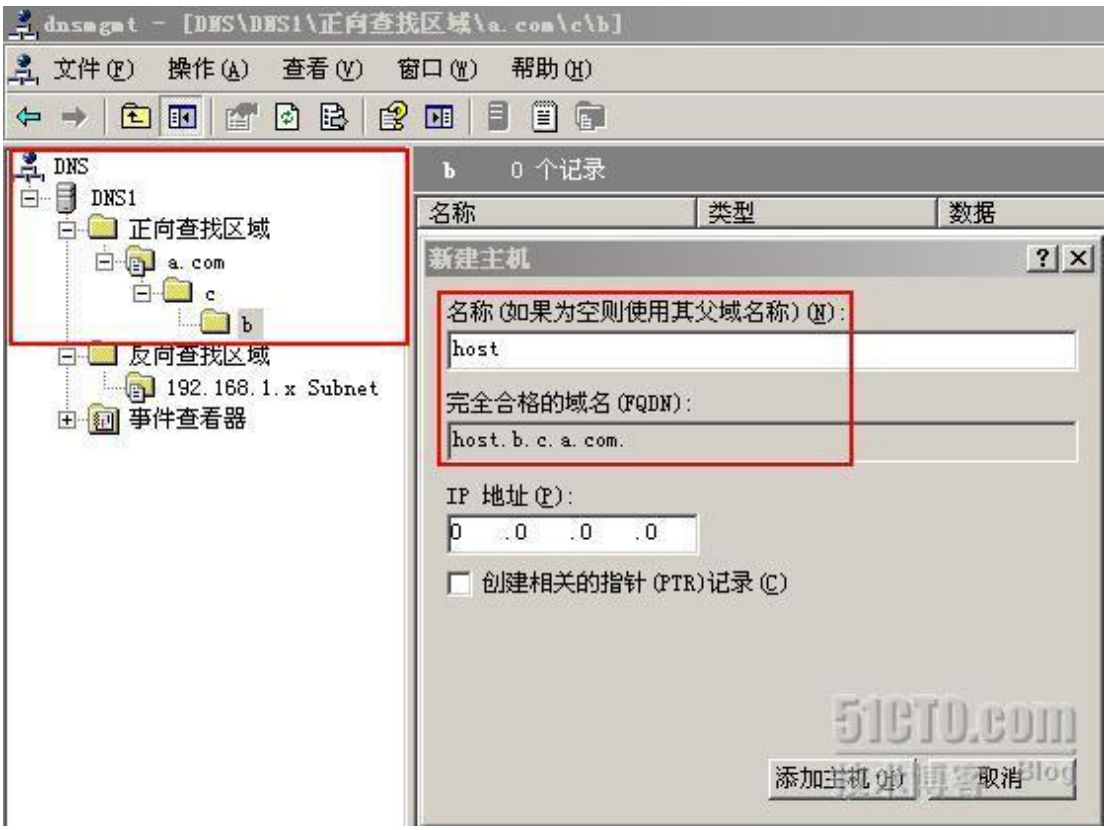
从上图得知，新建的 a.com 区域已有两条默认的记录，即 SOA 和 NS 记录。下节会重点讨论这部分内容。

这里我们右击 a.com，会发现只能新建【域】而并非区域。如下图：

因为区域和区域是相互独立的，不存在包含关系，一个区域里可以创建多个域，而且可以同时创建，比如同时创建 2 个域，名称分别为 b 和 c ，如下图：

创建多域的写法就是用句点符号将各个域名称隔开，请注意先后顺序，其实隐含的，这个顺序也体现了 DNS 域名空间的层级顺序。我们可以从下面的图示中得到印证。此处点击【确定】后，如下图：

上图中可以看到自动创建了 2 个域，且是包含关系。我们选择 b 域，并打开新建主机 A 记录的窗口，如下图：



图片看不清楚？请点击这里查看原图（大图）。

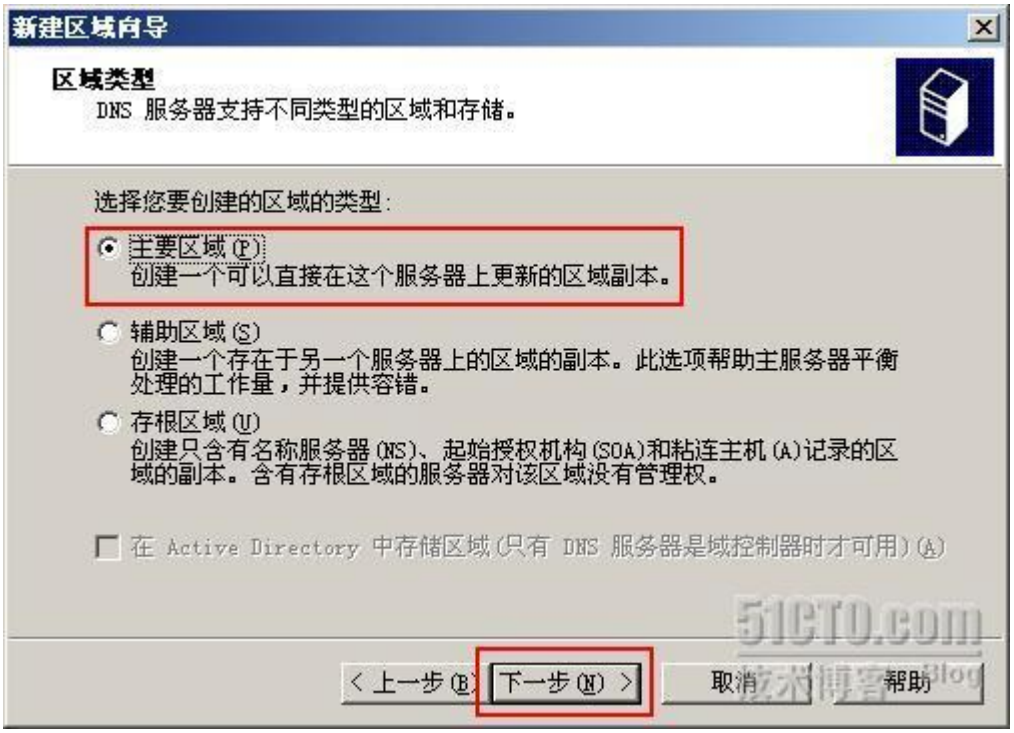
当创建一个主机名为 host 的 A 记录时候，完整的名称是 host.b.c.a.com. 请注意顺序。请大家留意，在此名称最后的那个. ，其实就是整个域名体系的根域。

正向查找区域创建完成后，才可以创建反向查找区域，从字面上就不难理解，所谓反向查找，也就是根据 IP 来反查对应的名称记录。现在演示一下这个创建过程。

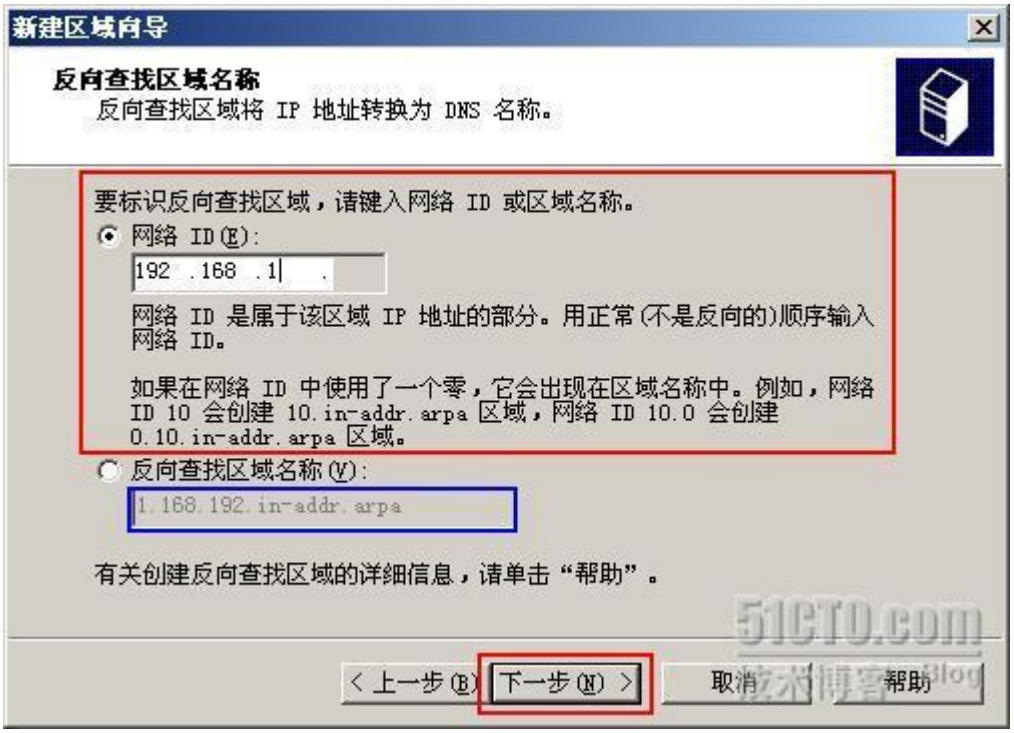
右键点选【反向查找区域】，并选择【新建区域】，如下图：

依照上图选定后，出现如下图示：

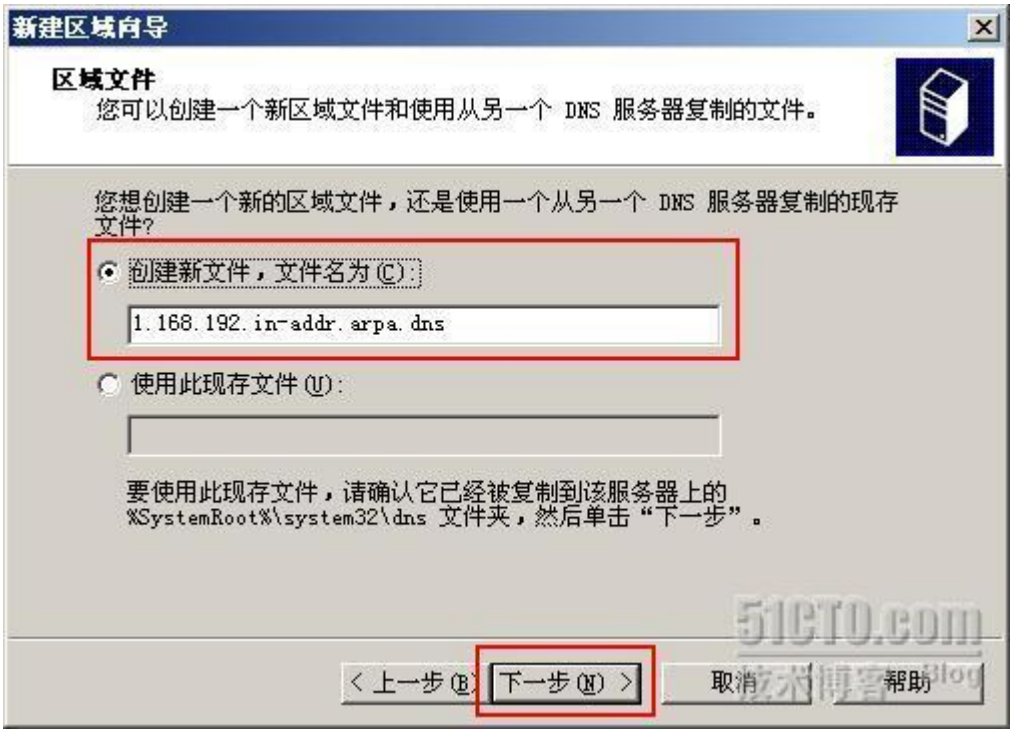
和创建正向区域类似，我们直接点击【下一步】继续，如下图：



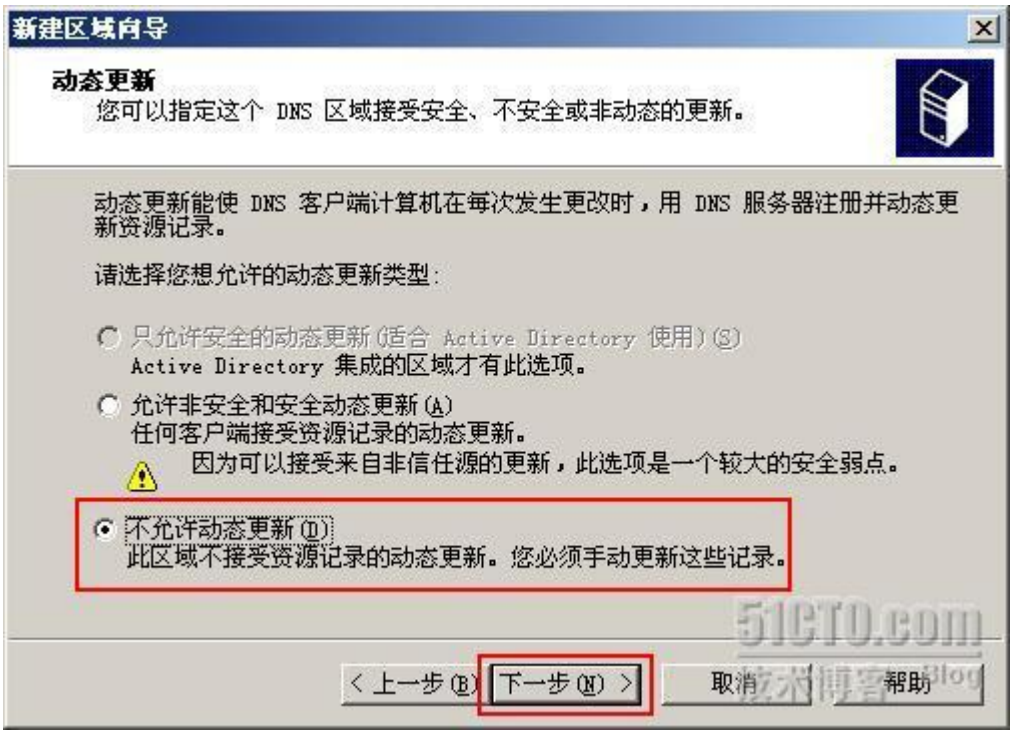
由于 a.com 是一个主要区域，如果要建立这个区域的反向查找区域，就需要区域类型一致，所以这里选择【主要区域】并点击【下一步】继续，如下图：



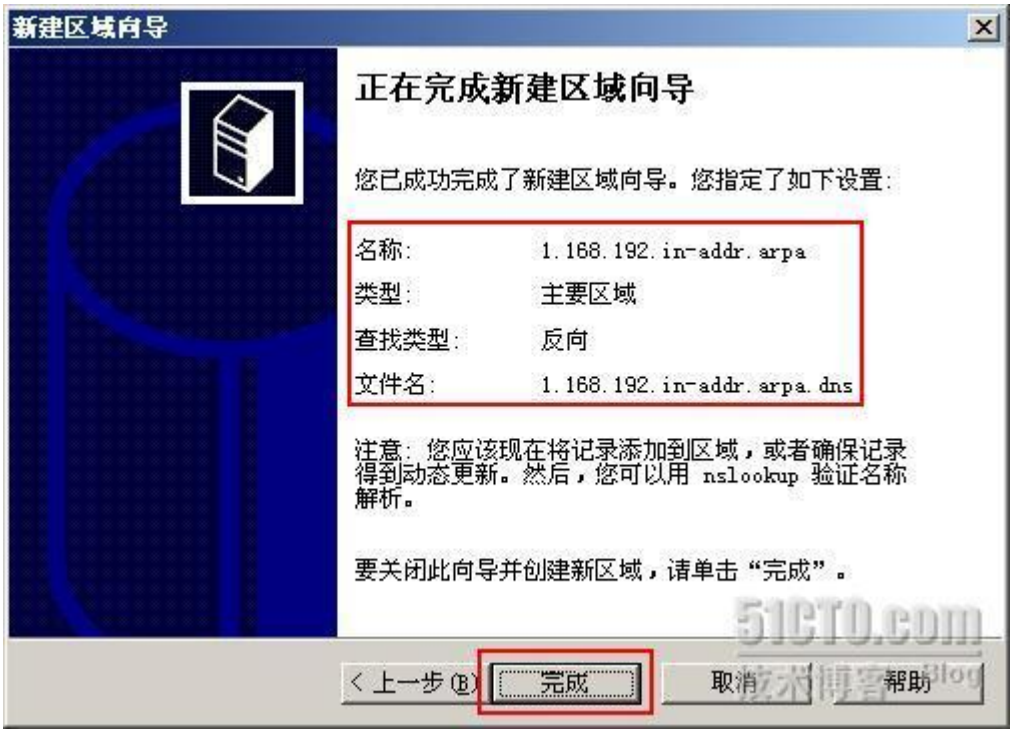
此处填入的内容是根据 DNS 服务器的 IP 和掩码进行与运算得到的一个网络 ID，由于本机 IP 是 192.168.1.1，且使用了标准 C 类地址的掩码，所以网络 ID 应该是 192.168.1，依此填入空白处，同时在蓝色框内会自动产生对应的反向区域的名称：1.168.192.in-addr.arpa，这个名称正好是网络 ID 的反向格式。填写完毕后点击【下一步】继续，如下图：



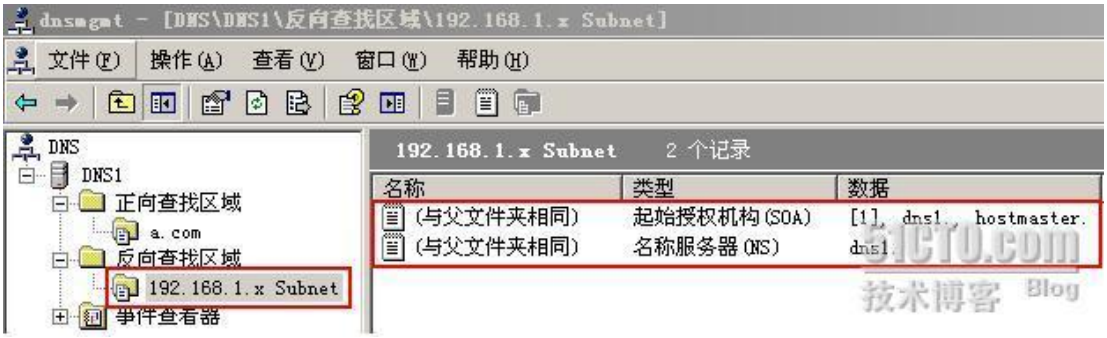
和正向查询一样，反向查询区域的信息也会以文本格式保存在本机中，此处默认即可。点击【下一步】继续，如下图：



这里的设置和正向区域一样，点击【下一步】继续，如下图：



在向导的最后，也显示了所创建的反向区域的一些基本信息，点击【完成】后结束此向导。打开这个反向区域，如下图：



图片看不清楚？请点击这里查看原图（大图）。

这里也可以看到 SOA 和 NS 记录，这部分内容和正向区域的如出一辙。

下节，会着重讨论正向查找区域（反向亦同）的属性的各个概念和知识点，内容很多也很重要，敬请期待！

谢谢大家的支持！

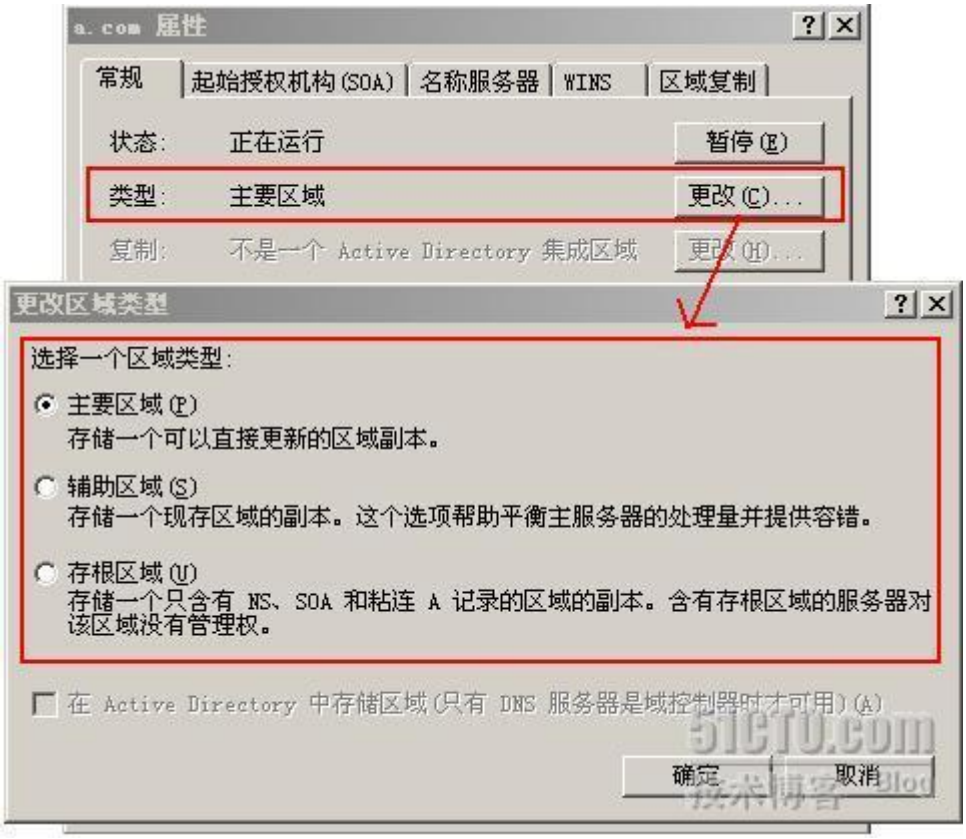
一起学 DNS 系列（六）详解正向、反向查找区域

上节演示了正向和反向区域的创建过程，本节开始着重讨论 2 者的属性，即区域属性，我们将以正向区域的属性为主，当然两者属性并非完全一样，区别之处会另作讲解。

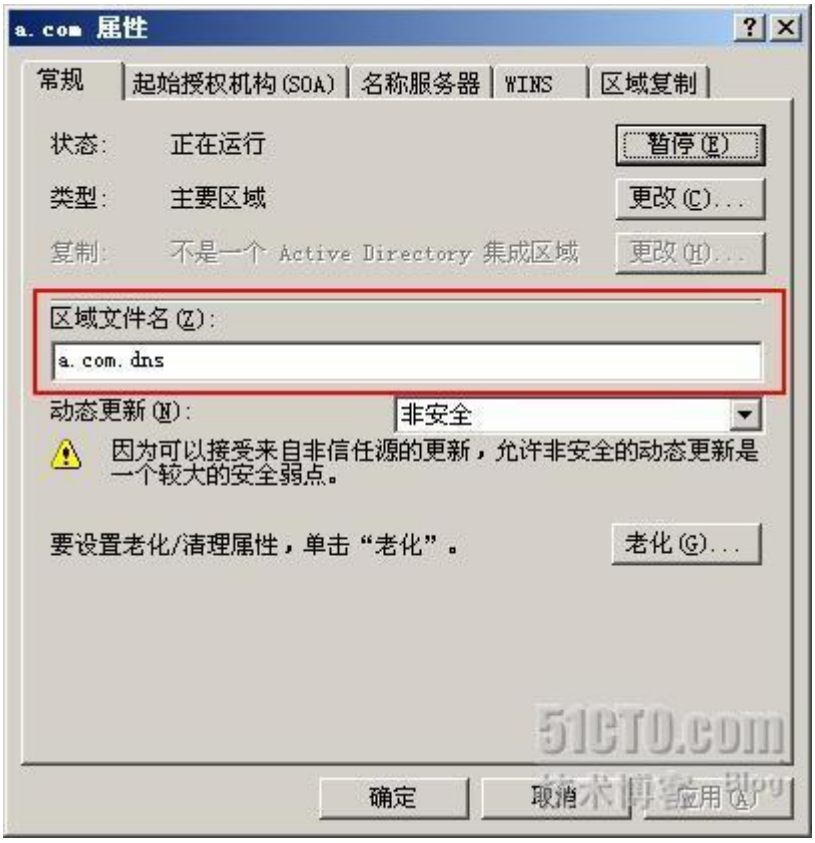
正向查找区域（Forward Lookup Zone）

打开 a.com 区域的属性，如下图：

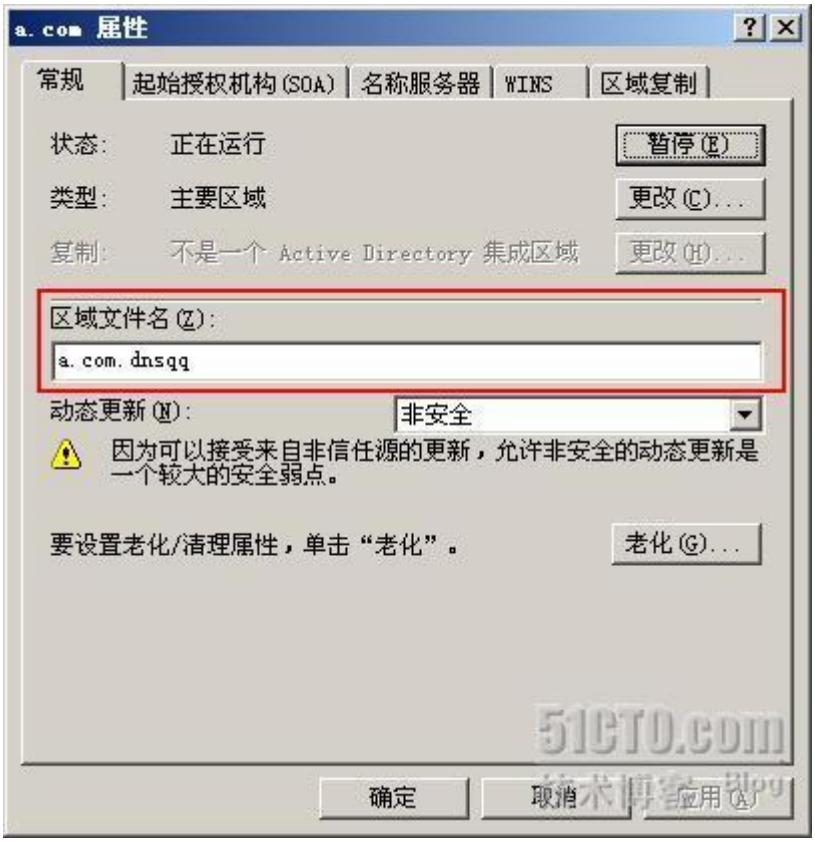
首先是【常规】选项卡，这也算是 Windows 的一个特色了，很多设置界面的第一个选项卡通常都是这样的。在这个选项卡里，我们需要关注以下几个属性。首先，我们可以修改当前区域的类型，如下图：



在这里选项里，我们可以把当前的 a.com 这个区域的类型改成辅助区域或存根区域，这个设置比较重要，所以下一节将讲解和演示辅助区域和存根区域的操作过程。这几个区域类型的基本概念在上一节里已有所提及，这里就不再重复了。一般情况下，我们在创建区域的初期就需要规划好区域的类型，减少改动的频次，同时也可以避免不少麻烦的产生，这里我们使用默认值即可。最后一个设置是灰色的，是因为此区域并非建立在域环境下所以无法选择。同样在第一个图中的复制也不可选，工作组环境下无法将区域设置集成到活动目录中。再来看一下区域文件名，如下图：



默认的区域文件名的名称由两部分组成，即【区域名称】和字符【dns】，中间用一个句点衔接。当然，我们也可以在这个设置里直接更改，比如在名称后面加一些字符，如下图：

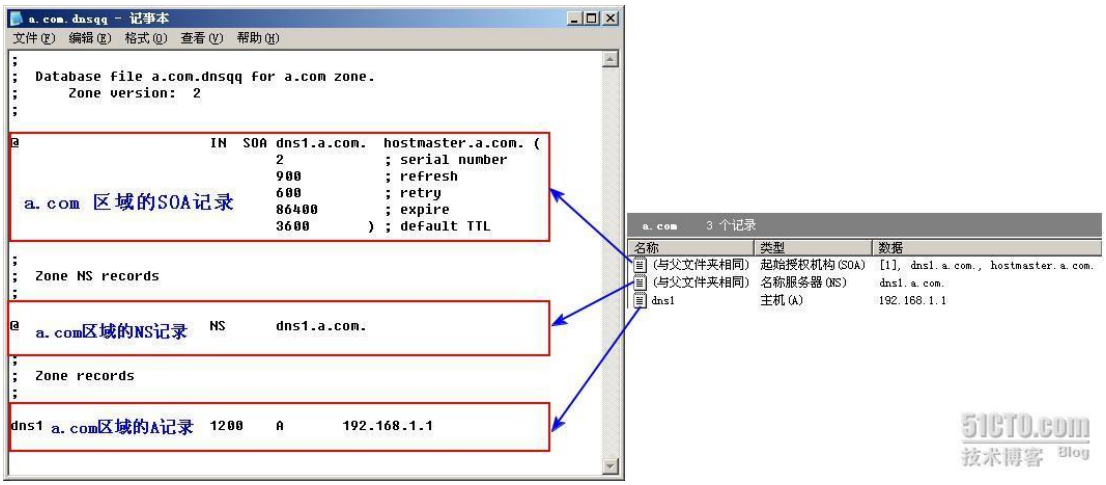


可以看到我已经修改了区域文件名，那原来的 a.com.dns 呢？被覆盖了吗？其实并非如此，我们可以到 c:\windows\system\dns 目录里下看一下 DNS 的系统文件。如下图：

图片看不清楚？请点击这里查看原图（大图）。

从上图得知，原来的文件并没有丢失，改变名称对其的影响只是将原来的文件复制一份，并把名称变成了 a.com.dnsqq，内容还是一样的。而且请大家注意，这个文件的文件类型是

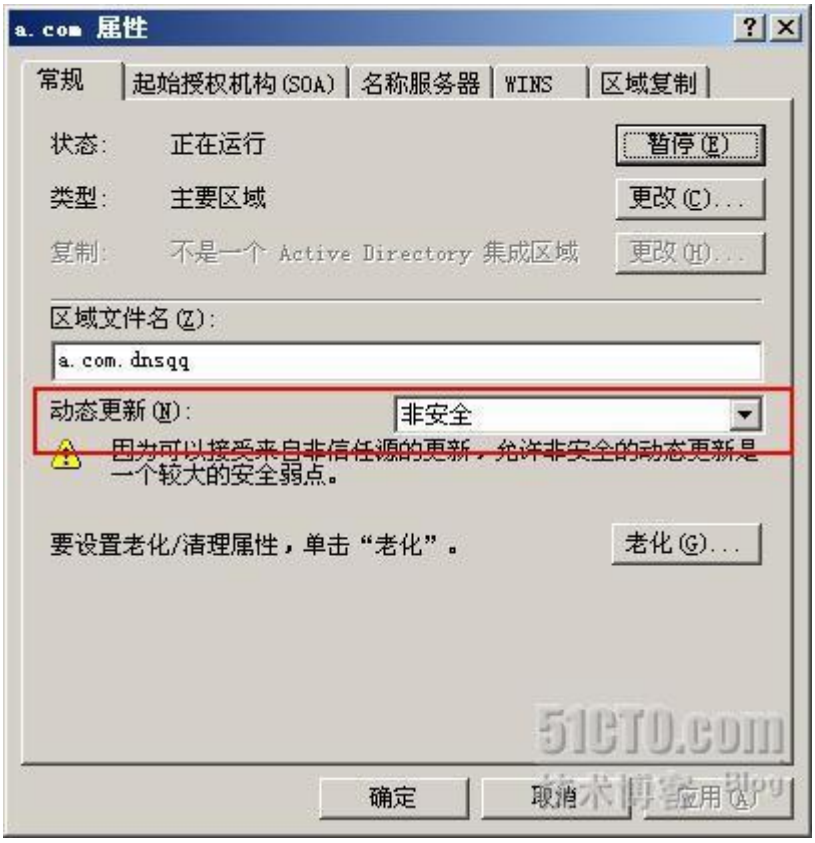
DNSQQ 文件，是以后半部分名称来命名的。可见，此文件是一类无扩展名的文件类型。我们利用 notepad 打开这个配置文件，看看里面记录了什么信息。如下图：



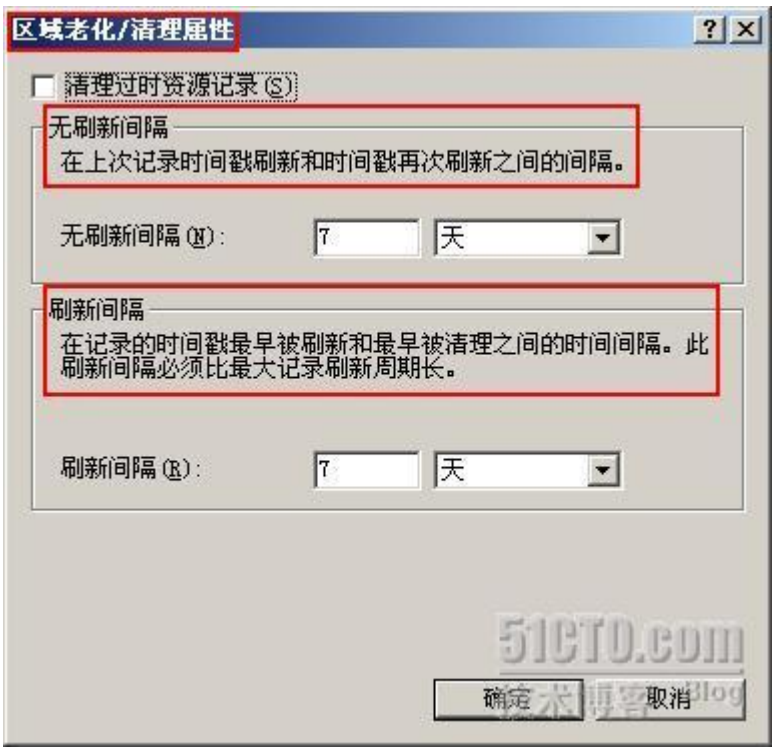
图片看不清楚？请点击这里查看原图（大图）。

在这个图中，我们可以把文件分为三部分，具体内容已经用蓝色字标明，其实这部分内容也就是这个区域的数据的体现，这个正好可以和 DNS 管理器下的 3 个记录一一对照，上图中的三个箭头也指明了各自的对应关系。其实，第一和第二部分内容也就是区域属性里的【起始授权机构】与【名称服务器】选项卡的内容。

在区域名称下方，我们可以设置这个区域是否允许动态更新，如下图：



所谓动态更新，也就是说当客户机的 IP 或主机名发生变化时，对应的 A 记录或其他记录是否会自动更新，在工作组模式下，只能选择【无】或【非安全】两种类型，因为这种模式无法对客户端的身份进行验证，但在域环境下就可以实现【安全】的自动更新，关于这部分内容会在第八节里详细说明。【常规】页的最后一个选项是【老化】，点选后如下图：



此选项是针对本区域内的记录设置的老化时间，当然我们也可以针对整个 DNS 服务器，也就是所有区域进行相应设置。当我们启用老化设置后，系统会创建一个时间戳，当 DHCP 客户端或服务器为 a.com 区域里的 A 记录进行动态更新时，会更新这个时间戳，也就是我们设置老化选项的时间点。这里要提一点，手工创建的记录的时间戳为 0，即不会老化。

图中的两个选项，【无刷新间隔】意为以上一次时间戳为起点，在一段时间内拒绝重复刷新时间戳，也就是说如果设置为 7 天，上一次刷新是在 1 月 10 号的零点，那么在 7 天内，及时有刷新，系统也不会更新这个时间点，依然是 1 月 10 号的零点。

在【无刷新间隔】后就是【刷新间隔】，这个时间设置意义在于，在某个时间段内，允许 DNS 客户端刷新资源记录且记录不会被清除掉。反之，如果超过了【无刷新间隔】和【刷新间隔】之后，没有被 DNS 客户端刷新，那么这些记录将会被 DNS 服务器清掉。通常，设置【刷新间隔】要大于或等于【无刷新间隔】。依据这些理论基础，如果启用上述的老化选项，当 DNS 客户端 14 天后依然没有被刷新，那么这些记录将被清掉。

我们来看下一个选项卡，即【起始授权机构 SOA】。如下图：

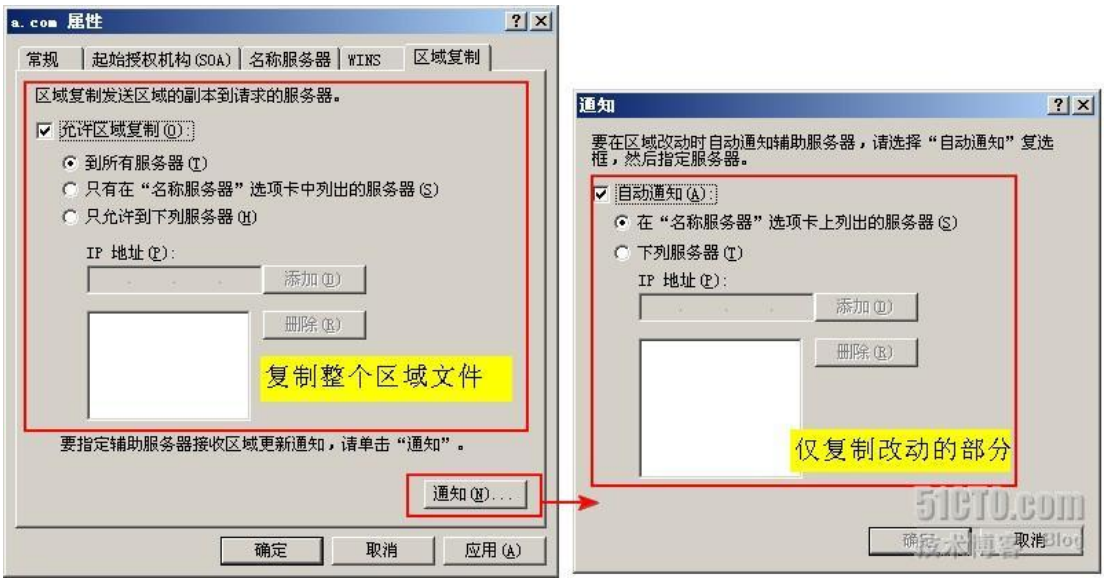
所谓 SOA 记录，即起始授权机构，字面上看有些不易理解，其实我们可以从个两方面来形象化理解 SOA。首先对于服务器而言，SOA 记录类似一个基本的基础数据记录点，当 DNS 服务器启动时会读取 SOA 记录，以确定一些关键信息，比如指派给此区域的 DNS 服务器的主机名称、以及负责该区域的人员名称等，这 2 个属性我们都可以手工进行添加。另一方面，当 DNS 客户端在更改或刷新本机的网络地址、主机名称时候，都会向自身的 DNS 服务器发起一个针对 SOA 记录的标准查询，目的是向 DNS 更新自己的记录，此时，如果 DNS 配置了自动更新，则会刷新本机的 DNS 记录。此后会有单独分析和讲解 DNS 相关命令的章节，敬请关注！

在这个属性页中，有一个比较关键的属性即序列号，它的用途是在使用到区域传输时，依靠序列号的大小来判断 DNS 区域文件版本的新旧。当修改区域名称、增加或删除记录，或者手动对版本号进行调整时，版本号都会发生变化。此时，如果我们在另一台 DNS 服务器上创建了本区域的辅助区域，则辅助区域会在固定间隔时间内查询主 DNS 服务器并获取

序列号，以确定主 DNS 的区域版本号是否大于自己的序列号，若大于，则会向主 DNS 发起区域复制。而这个固定时间就是上图中第二个红框所框选的内容。具体如下：

所谓刷新间隔，也就是辅助区域查询主区域以获取更新内容的时间周期。重试间隔生效的基础是在刷新间隔后区域复制失败，辅助区域重试时需要等待的时间，默认为 10 分钟。过期时间则相对较长，因为这个是当辅助 DNS 无法联系到主 DNS 时，允许利用自身的区域信息来答复 DNS 客户端请求的时间，超过此时间，辅助服务器则会将自身的本地数据当作不可靠数据。最后一个参数是最小（默认）TTL，此记录是应用到此区域内所有资源记录的生存时间，当 TTL 过期时，该 DNS 服务器将丢弃此记录的缓存。在最下方的 TTL 值是 SOA 记录的 TTL 值，当同时设置着连个 TTL 时，SOA 记录的 TTL 值将覆盖最小（默认）TTL 数值。

回过头来看，这一部分讲到了 2 个易混淆的概念，区域传输和区域复制。两者相互影响，区域复制是区域传输实现的前提，我们所说的区域传输，主要指的是 2 部分，1、将一个区域文件传送给一台或多台服务器；2、区域文件从主 DNS 区域传输给它的辅助 DNS 区域。如果不开启区域复制的功能，则主、辅 DNS 间无法实现区域文件的传输。而在区域复制时，我们还可以对这个复制进行更细化的设置，比如可以允许全部复制，还可以只复制更新的部分，这些都是出于优化和利用服务器资源的目的。这部分内容的设置是在最后一个选项卡【区域复制】中，说到了区域的传输，所以这就顺带把这个也提一下，设置界面如下图：

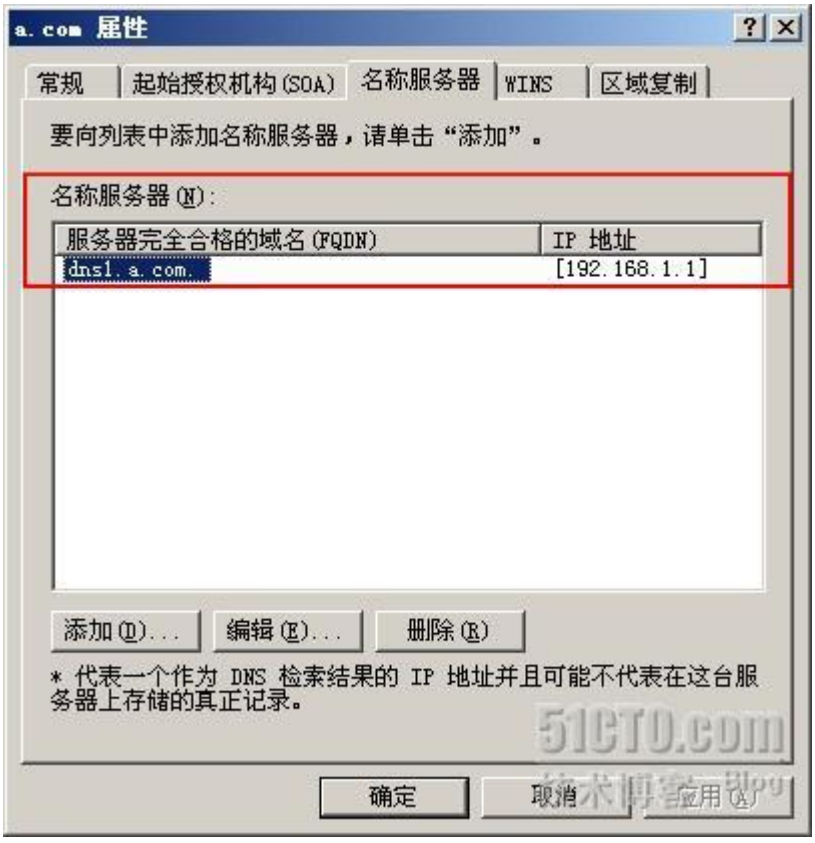


图片看不清楚？请点击这里查看原图（大图）。

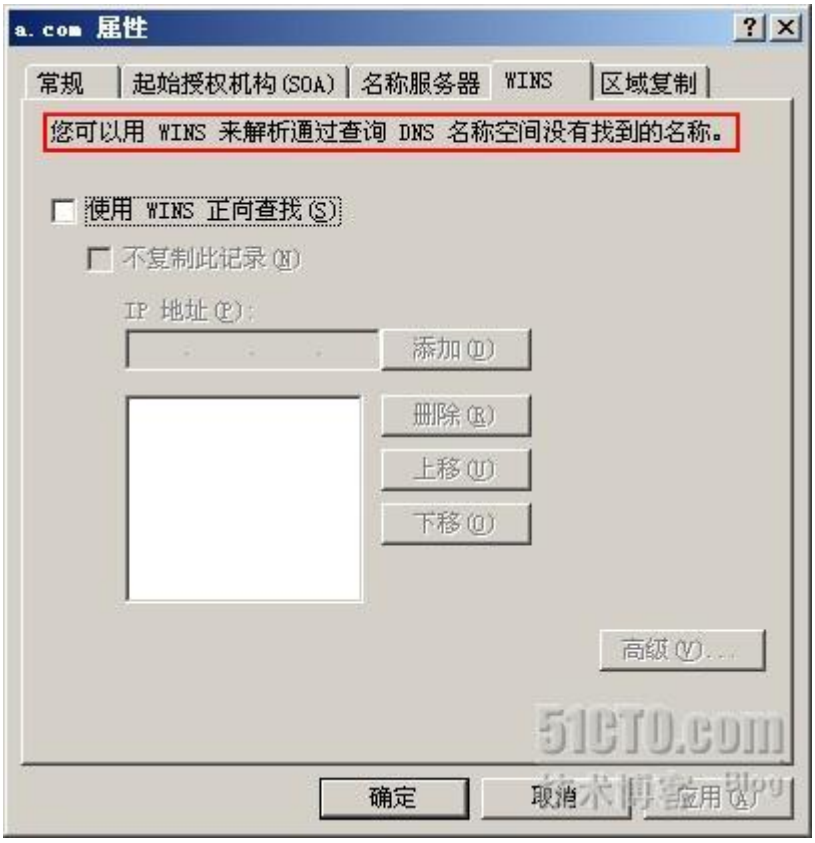
这部分内容比较好理解，我们可以限定的复制范围，比如到部分或所有服务器。

我们再回过头来看剩余的两个选项卡，【名称服务器】和【WINS】。

首先看一下【名称服务器】，如下图：



名称服务器里会列出本区域内的主 DNS 服务器，目的在于指定此 DNS 区域中的权威 DNS 服务器是那一台。当然，我们也手工可以进行添加。下图是最后一个属性：



这个设置的目的是当DNS无法解析某一主机的FQDN时，DNS会使用已配置的WINS，并通过查询 WINS 的 NETBIOS 名称空间记录来辅助查找对应 FQDN 的主机名，相当于多了一种解析的方式。不过现在我们很少去用到 WINS 来辅助 DNS 查找名词，所以通常这一项都是默认不需要设置。

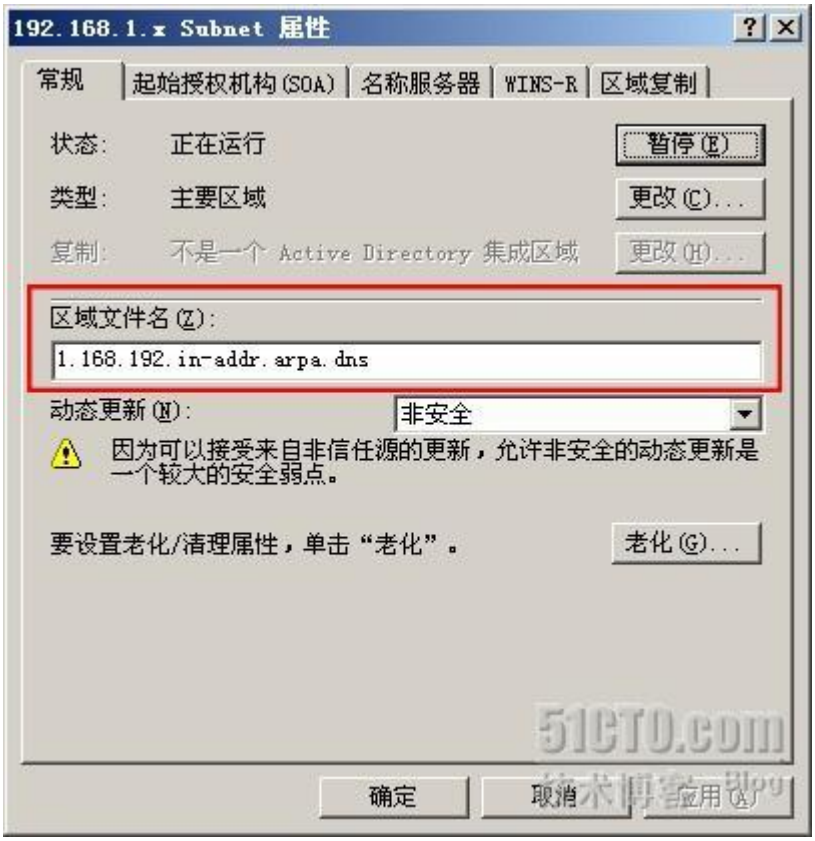
反向查找区域（Reverse Lookup Zone）

相对于正向查找，反向查找区域用的相对较少，这个功能可以允许客户端通过查询 IP 地址得到对应的名称，反向查询的类型我们称之为 PTR（Poniter），所以反向查询也被称为指针查询。在 DNS 系统里，一个反向地址对应一个 PTR 记录（与 A 记录相对应）。反向查询的整个结构和整个 DNS 域树结构相似，但不同的是根节点不是单纯的一个【.】，而是.in-addr.arpa. ，这部分是固定不变的。

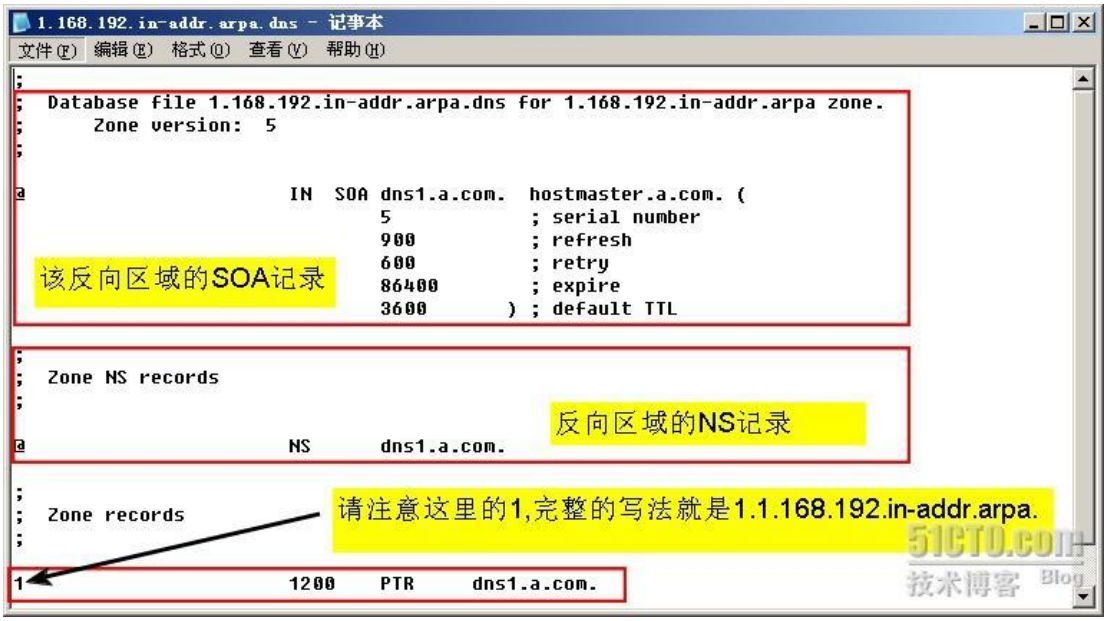
之所以需要设置这样一个域来实现反向解析，主要是考虑到如果按照正向解析的结果进行反查，那么当 DNS 名称空间异常庞大时，遍历整个空间来查询某一个 IP 对应的计算机

名称时将会异常缓慢，从而影响整个 DNS 名称空间的解析性能。因此在 DNS 标准中就定义一个特殊的域，即 in-addr.arpa，对应的子域则是反向构造的点分十进制的 IP 地址。也就是说当需要添加新的 PTR 记录时，只需要将对应的 IP 地址倒置即可。比如 IP 为 192.168.1.2 的反向记录，即表示为 2.1.168.192.in-addr.arpa。

上面说了有关反向区域的一些知识点，由于正、反区域的属性选项很类似，这里只说一下不同之处。由于反向区域的特殊性，因此对应的区域文件也与正向区域有 2 处不尽相同。如下图：

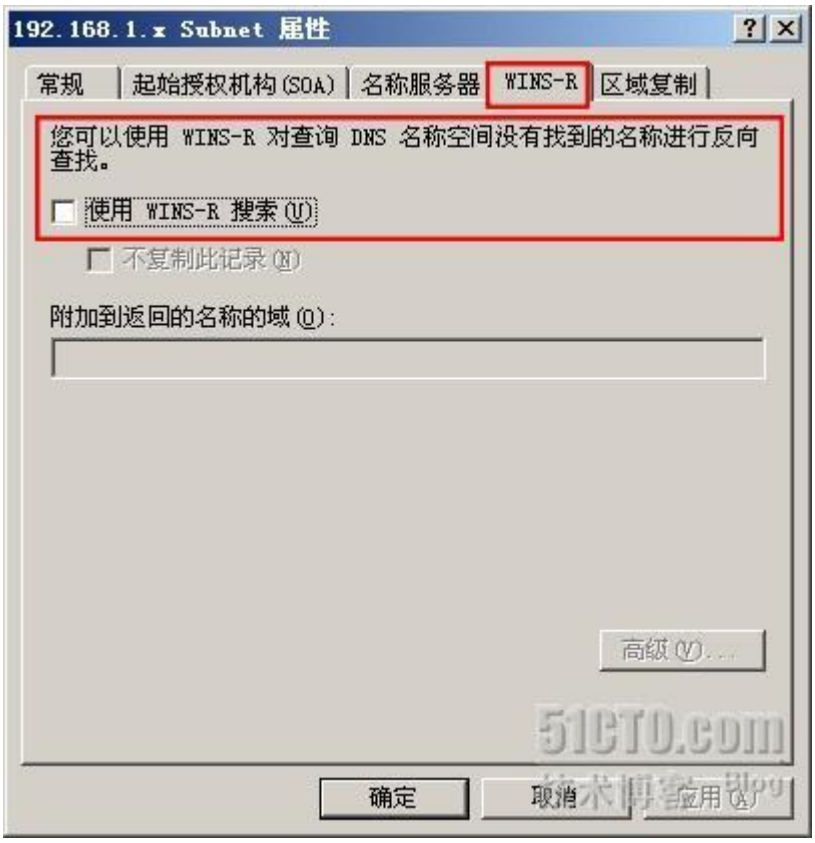


红框中标记的就是 a.com 反向区域的文件，此区域内创建的 A 记录的 PTR 记录都会存在于此文件中。如下图：



图片看不清楚？请点击这里查看原图（大图）。

还有个区别就是在 WINS 辅助解析选项卡，如下图：



其实，WIN-R 就是 WINS 的反向解析，R 即为 Reverse 之意。

本文简要的分析了一下正向、反向区域的各个属性选项，可能有的知识点没有说的很详细，不过后面的章节仍会再次碰到这些内容。

下一节会演示辅助区域和存根区域的使用，敬请期待。

学 DNS 系列（七）辅助区域、存根区域操作演示 (1)

上节主要讨论了正向、反向查找区域的各个属性，今天起我们开始针对一些重要属性进行讲解和演示，工作中也会用的到，所以这部分内容比较重要。

我们知道，区域类型一共有三种，即主要区域 、辅助区域和存根区域，本节会针对辅助区域和存根区域的操作进行演示。

辅助区域操作演示

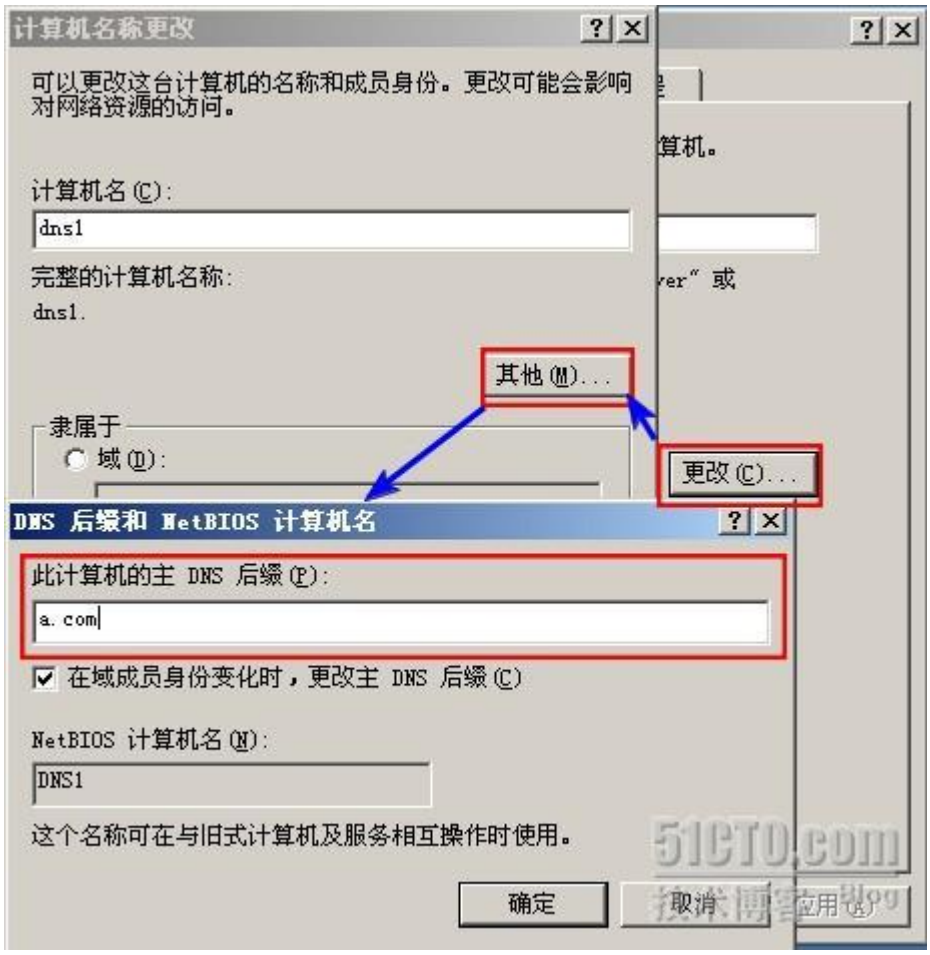
1、安装配置主 DNS 区域

辅助区域是以主要区域为蓝本，复制出一个相同，且可以为 DNS 客户端提供解析服务的副本区域，主要区域只能有一个，但我们可以为这个主要区域创建多个辅助区域。

这里准备了一个简单的试验环境，2 台 win2003 的服务器，安装 DNS 组件，分别创建主要和辅助的 DNS 区域，名称定为 a.com。实验拓扑图如下：

辅助 DNS 在查找主 DNS 时需要用到 SOA、NS 和 DNS1 的 A 记录，而默认情况下新建区域只会产生 SOA 和 NS 两条记录，所以我们需要为本机主 DNS 后缀，否则无法自动创建主机 DNS1 的 A 记录。

选择【我的电脑】属性后，具体操作如下图：



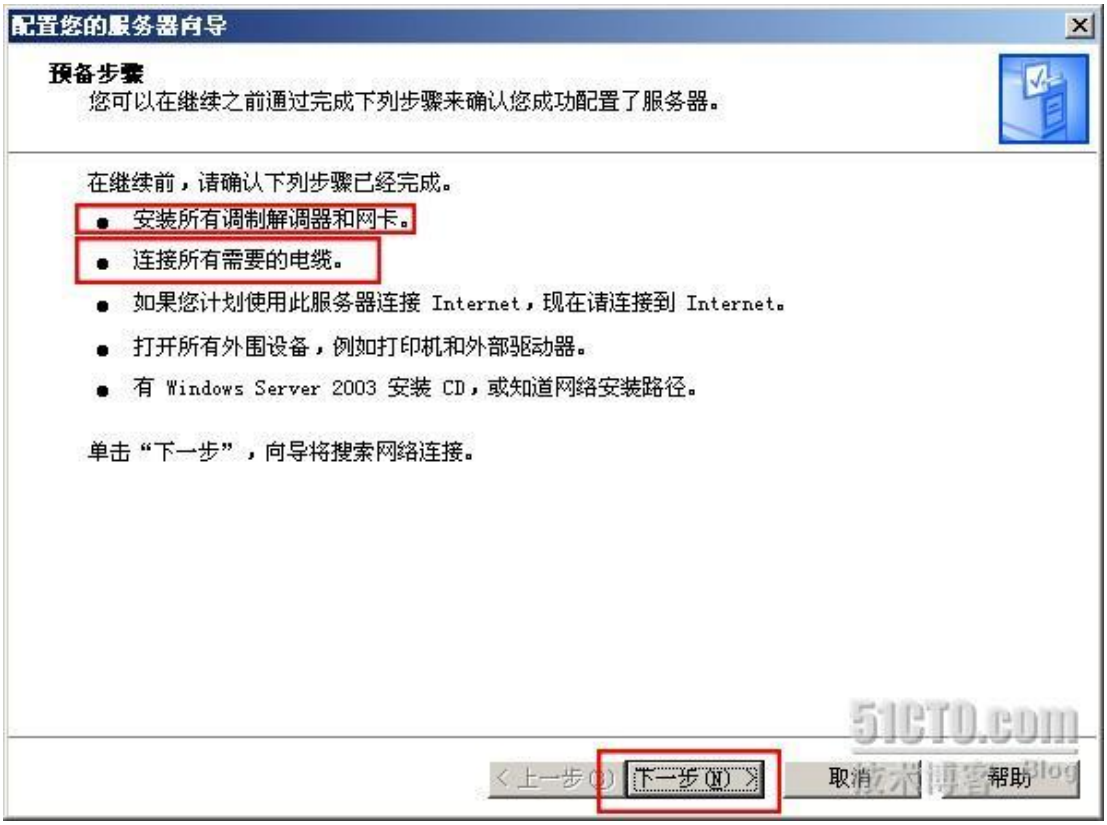
依照本系列第四节的内容，我们利用服务器配置向导来创建 a.com 区域，这样安装相比组件添加的方式配置更灵活多变。首先运行【管理您的服务器】，如下图：


Administrator
管理您的服务器
Windows 资源管理器
命令提示符
记事本
PicPick
Wireshark
Internet Explorer
SnapShot
所有程序(P)
我的电脑
控制面板(C)
管理工具
连接到(T)
打印机和传真
帮助和支持(H)
搜索(S)
运行(R)...
注销(L)
关机(U)
开始
C:\Documents and Se...

点选后，如下图：

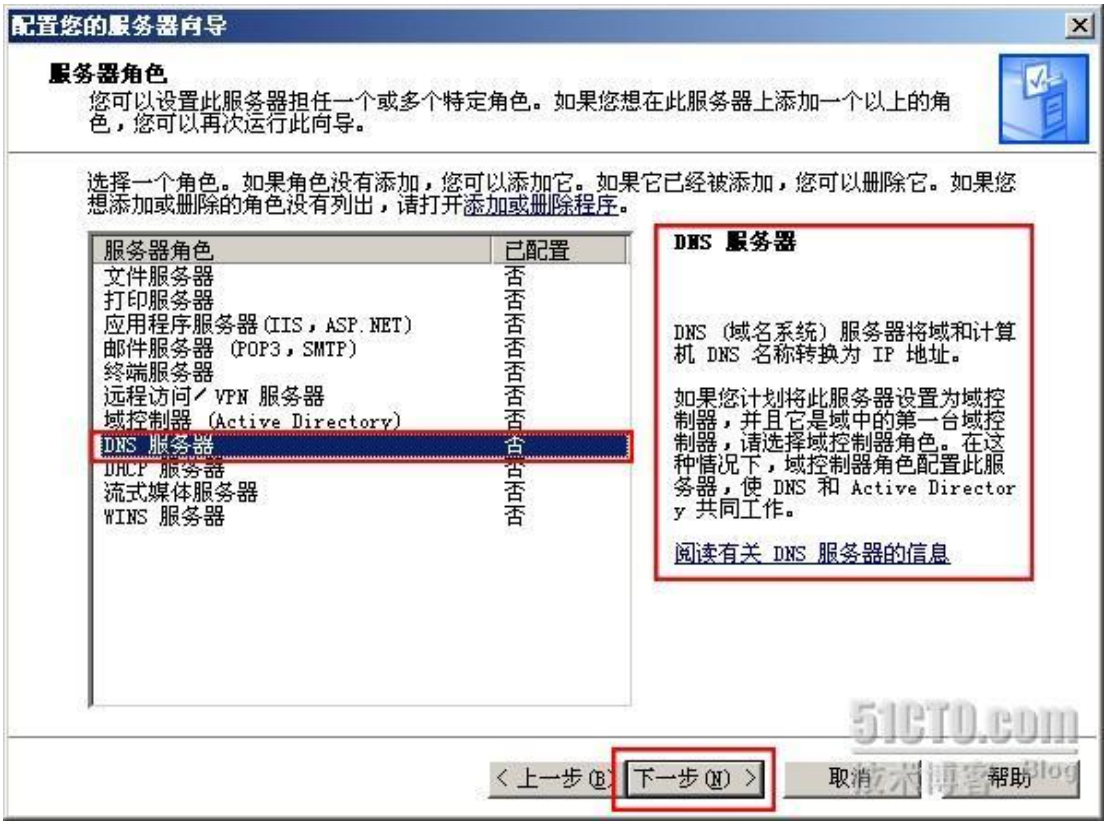
图片看不清楚？请点击这里查看原图（大图）。

此时服务器名称为 DNS1，我们选择【添加或删除角色】，如下图：



图片看不清楚？请点击这里查看原图（大图）。

请注意几个先决条件，网卡、网络等需要正常，且事先要为本机配置一个固定的 IP 地址，同时首选的 DNS 地址也应为自身。设置好后点击【下一步】继续，如下图：

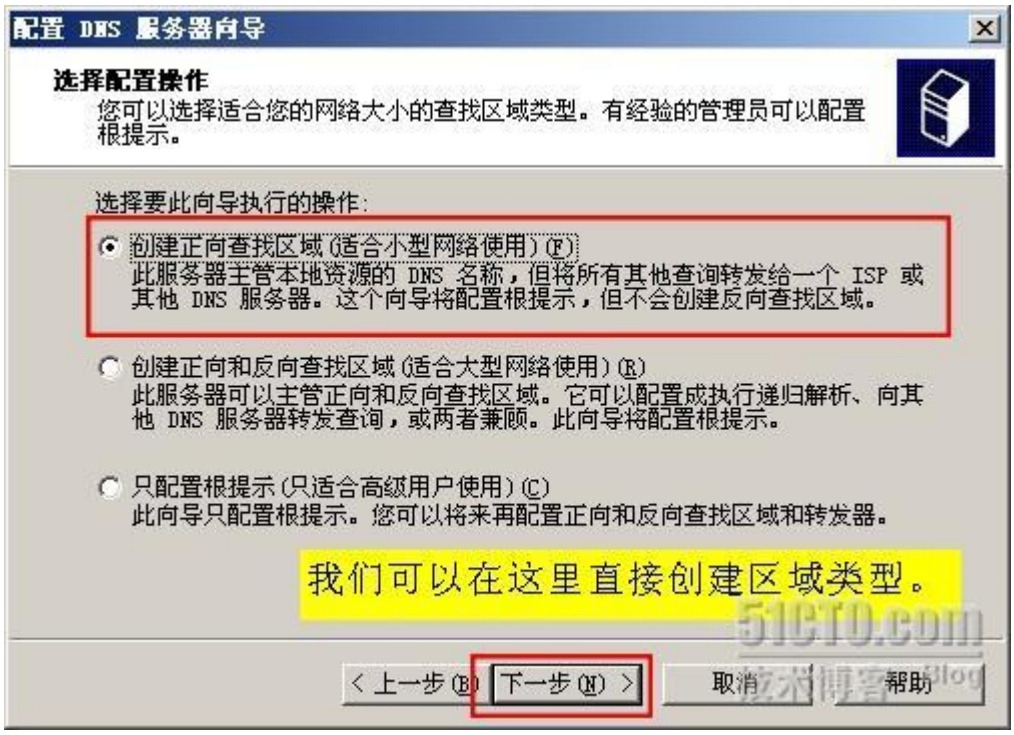


图片看不清楚？请点击这里查看原图（大图）。

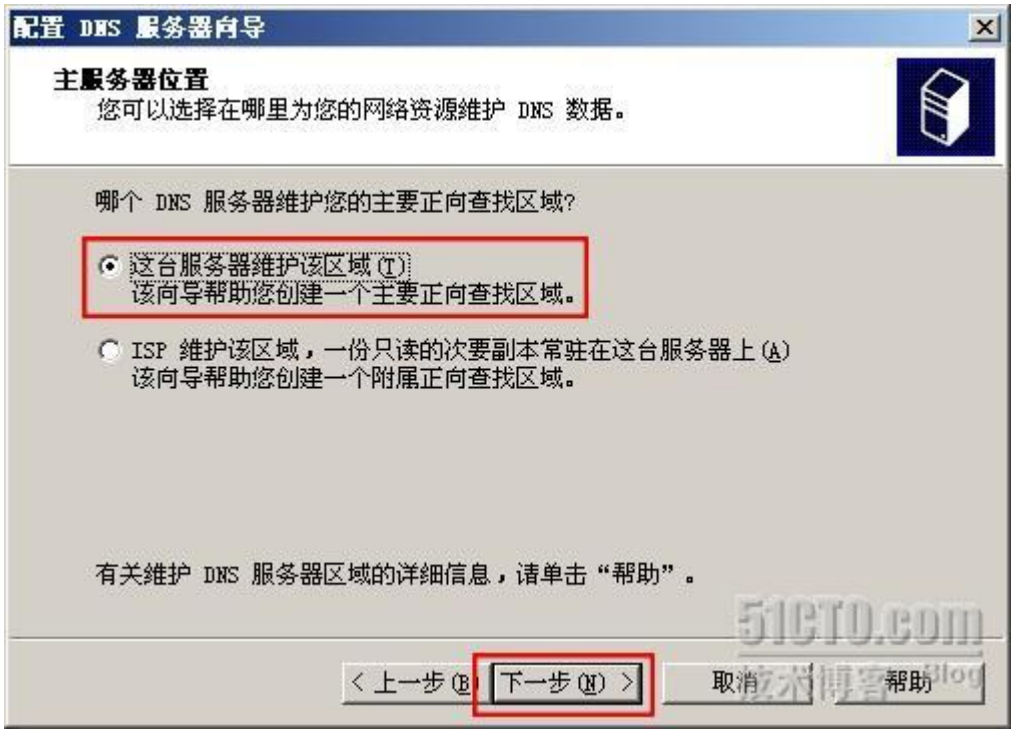
这里选定要安装的组件并点击【下一步】继续，如下图：

这是一个确认信息，点击【下一步】继续，如下图：

点击上面的【DNS 清单】可以查看 DNS 的部署清单，这里我们直接点击【下一步】继续。如下图：



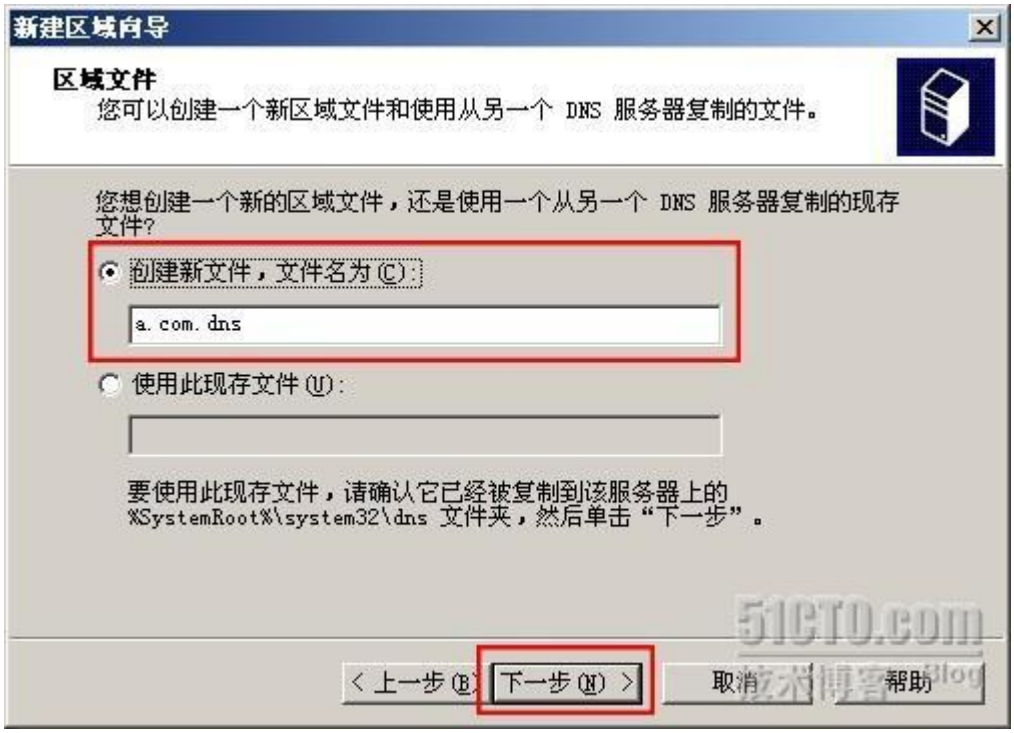
默认只创建正向查找区域，选定后并点击【下一步】继续，如下图：



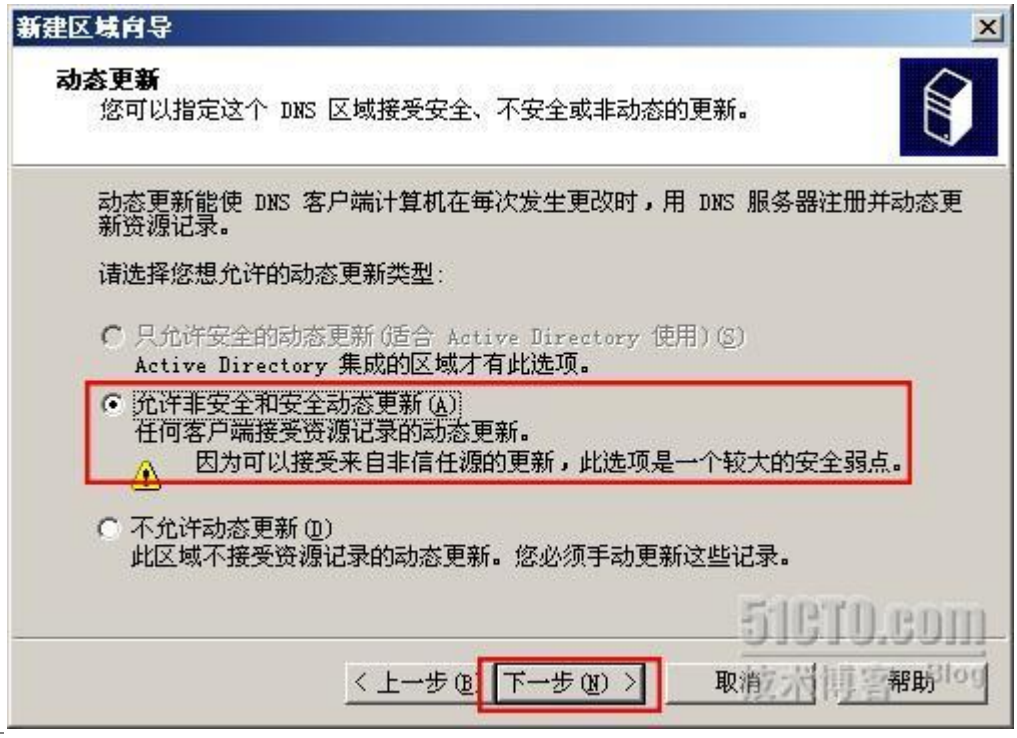
由于我们是在本机配置的 DNS 服务器，所以这里选择第一项。

选定后并点击【下一步】继续，如下图：

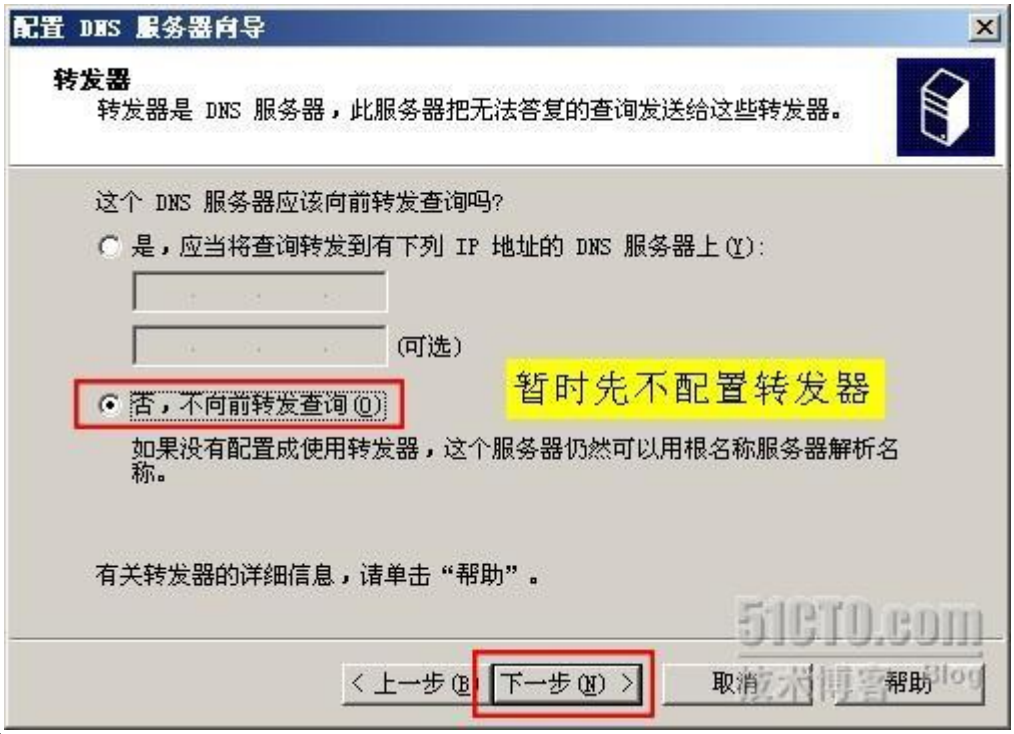
输入要创建的区域后并点击【下一步】继续，如下图：



这些步骤我们都很熟悉了，故不再熬述。点击【下一步】继续，如下图：



我们选择允许非安全和动态更新，选定后点击【下一步】继续，如下图：



此处先不配置转发器。选定后点击【下一步】继续，如下图：

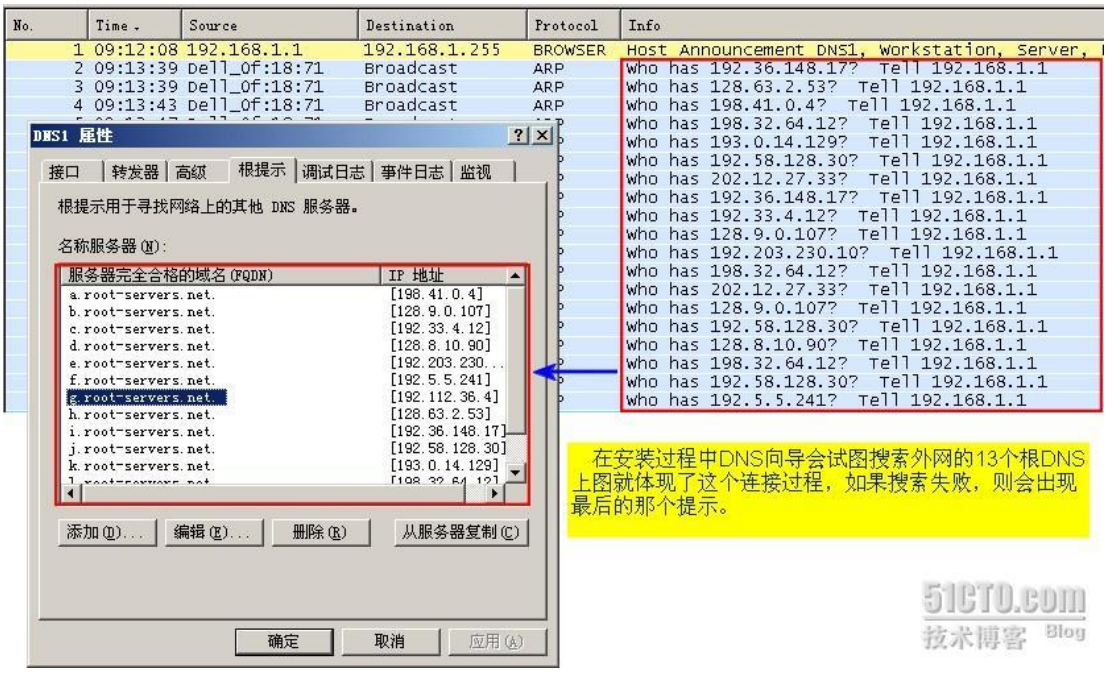
此时系统提示正在收集根提示，待完成后，会出现如下图示：

此图表明已经成功配置了正向查询区域 a.com。但当我们单击完成后，系统会弹出一个错误提示，如下图：

告诉我们“无法配置根提示”，点击【确定】后，也不会有什么异常。DNS 已经配置完成，那这个错误提示是什么意思呢？我们知道 DNS 解析过程中需要用到根提示，在创建 DNS

组件时，系统会自动去搜索网络上的 13 个公共的根 DNS 服务器，如果搜索过程失败，则会出现这个提示。反之，如果当前环境可以顺利联互联网，则不会出现这个提示。

为了验证，我们为这张网卡配置一个网关即 92.168.1.1，然后运行协议分析工具 wireshark 来观察此过程，如下图：



图片看不清楚？请点击这里查看原图（大图）。

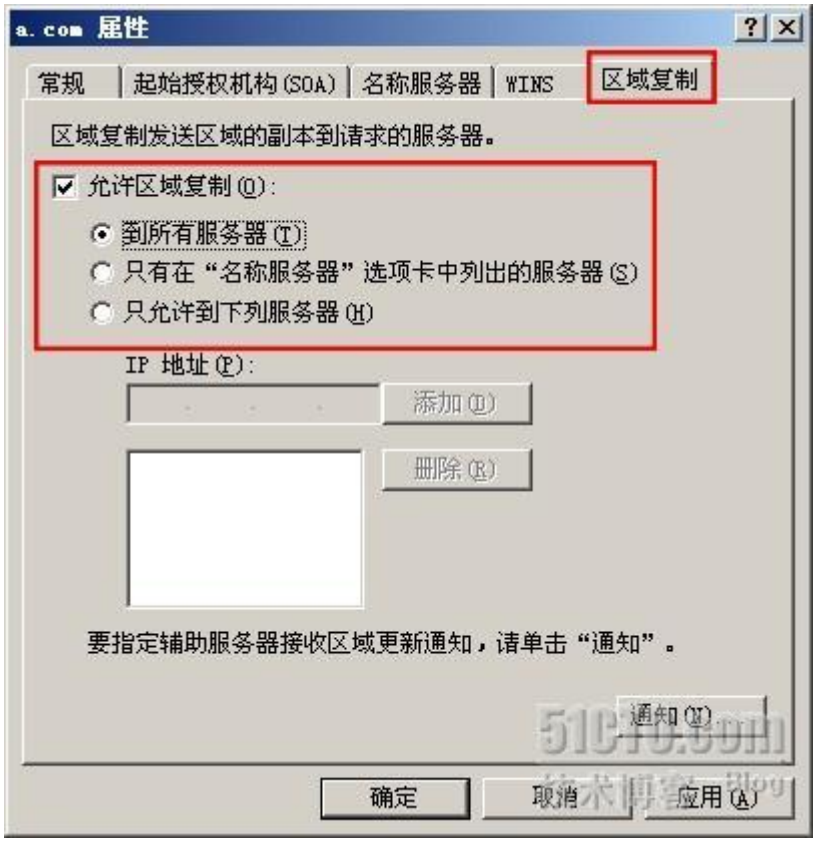
尽管是一个错误提示，但影响并不算大。完成后打开 DNS 管理器，如下图：

图片看不清楚？请点击这里查看原图（大图）。

区域创建完成后，自动产生了这三条记录，双击 DNS1 的 A 记录，如下图：

这里的 FQDN 也就是本机的完整计算机名。

在配置另一台 DNS 服务器前，我们还需要在主 DNS 服务器做些设置，也就是需要允许 a.com 区域被复制，这部分在上节有提到过。如下图：



在【区域复制】选项卡里，我们需要设置复制的目标对象，默认是只针对已列出的 NS 服务器，这里我们选的是【到所有服务器】，然后【应用】并【确定】即可。

到此，主 DNS 服务器已配置完成，下一小节将讨论辅助 DNS 区域的安装和配置。

敬请期待，谢谢！

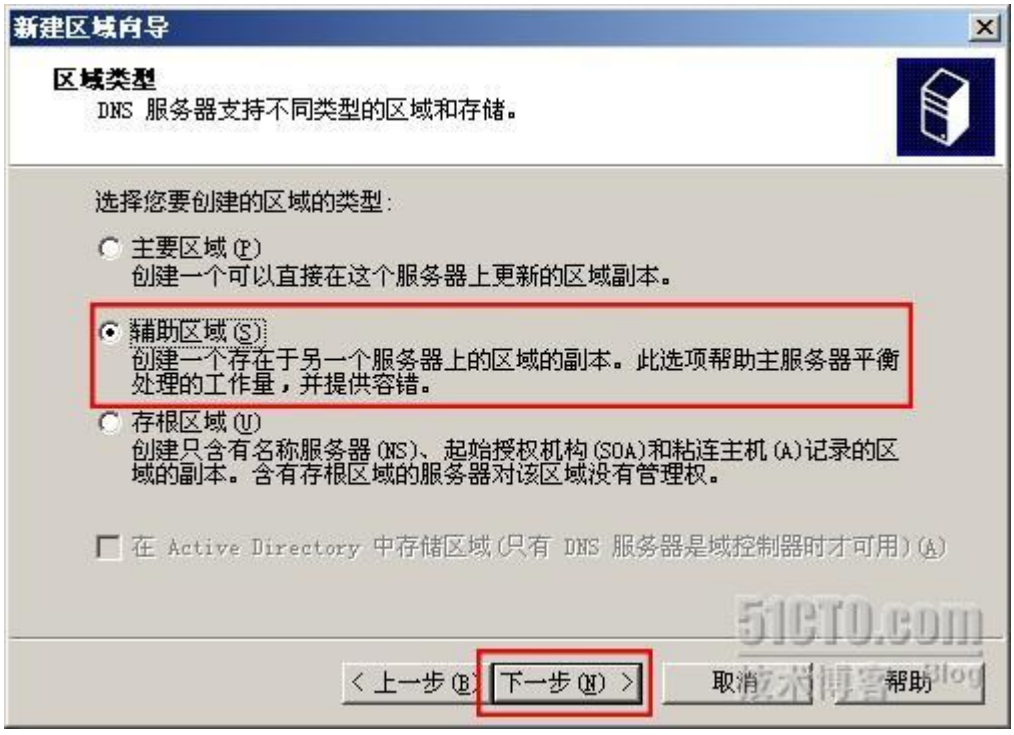
学 DNS 系列（七）辅助区域、存根区域操作演示 (2)

2、配置辅 DNS 区域

安装 DNS 的过程和上述类似，这里不再重复。下面开始创建辅助区域。打开 DNS 管理器，具体操作如下图：

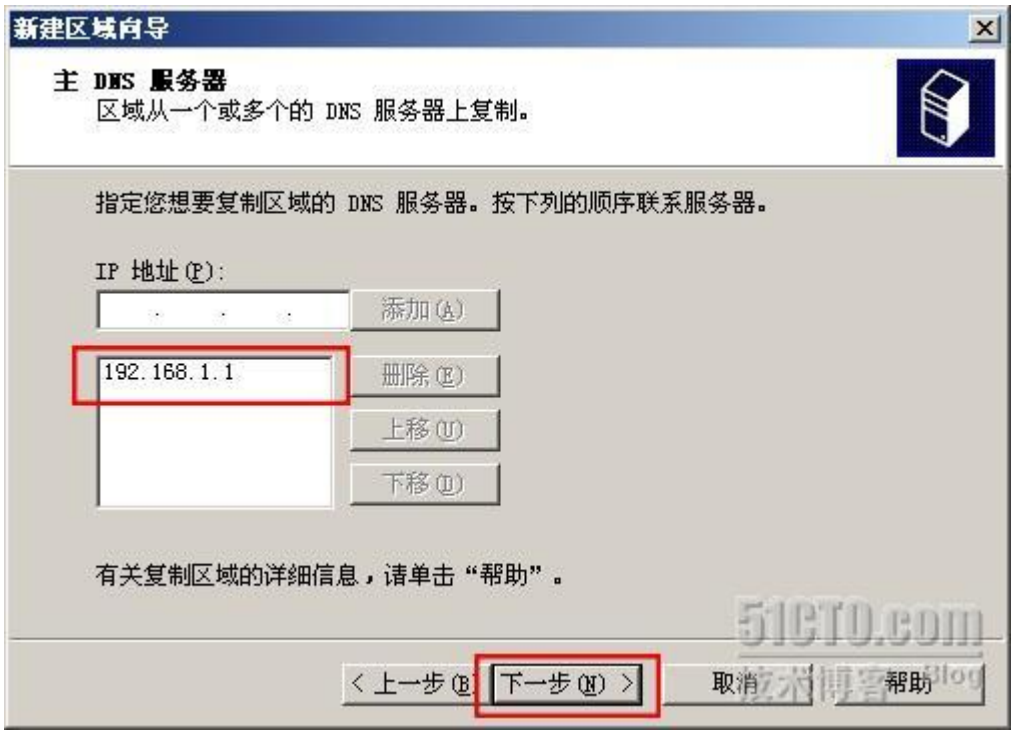
选择【新建区域】后，如下图：

直接【下一步】，如下图：

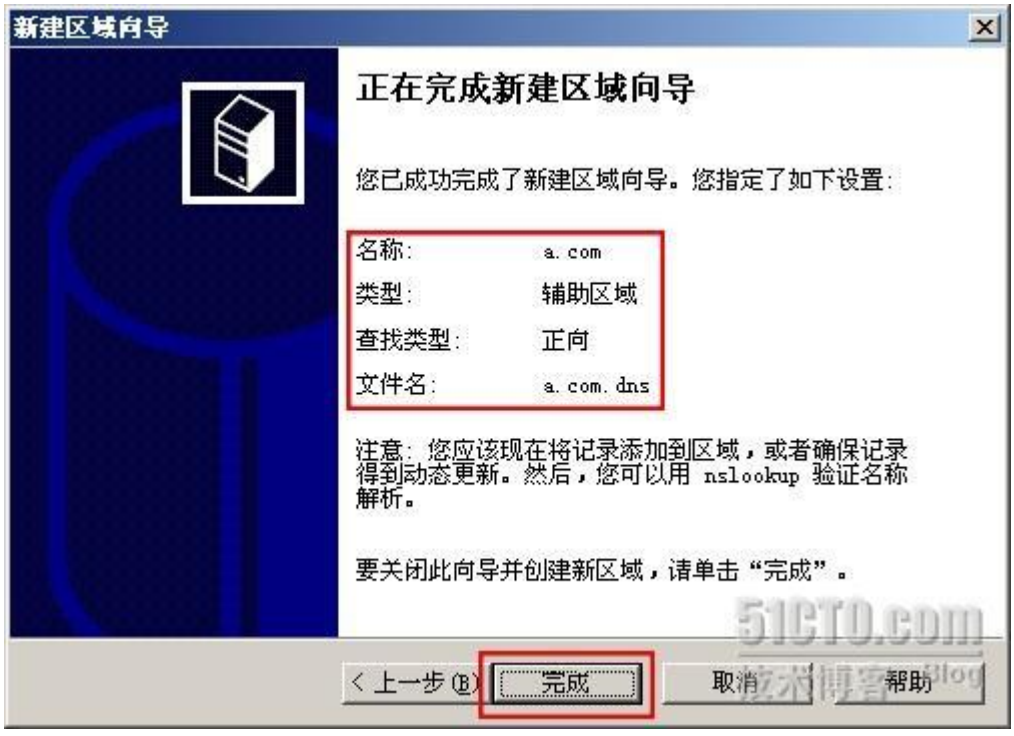


我们要建立主区域的辅助区域，所以这里的区域类型选择【辅助区域】。如下图：

这里输入的名称要和主区域的一致，设置完成后点击【下一步】，如下图：



此时向导提示我们输入主 DNS 服务器的 IP，设置完成后点击【下一步】，如下图：

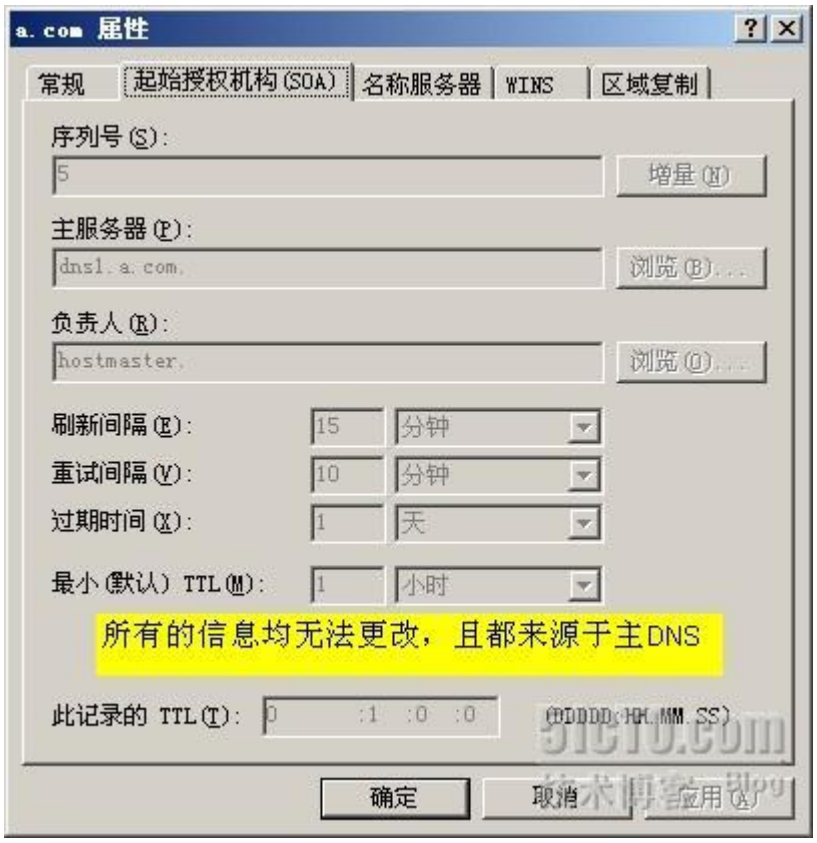


最后一步显示了设置信息，点击【完成】结束此向导。

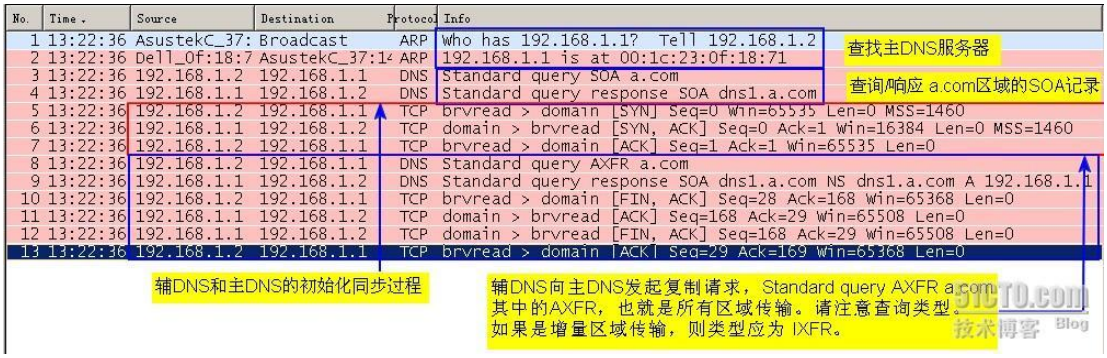
我们到 DNS2 上看看 a.com 的区域信息是否已被复制。如下图：

图片看不清楚？请点击这里查看原图（大图）。

可以看到在 DNS2 上也看到了和 DNS1 中一样的记录，双击 SOA 的记录，如下图：



设置都是灰色的，无法修改，表明这些信息是来源于主 DNS 服务器，NS 和 A 记录也是一样。复制过程耗时长短取决于区域记录的多少以及所采取的复制类型，即全区域传输，或是增量区域传输。默认是全区域传输，我这里也是采用的这种方式。由于记录很少，所以整个过程复制的很快。我们通过 wireshark 来看一下复制过程中，辅助区域和主区域间是如何通讯的。如下图：



图片看不清楚？请点击这里查看原图（大图）。

经过这个过程，辅助区域就建立完成了，那记录是如何更新的呢？

在 SOA 的属性里，有一个几个和时间有关的参数，那些参数就决定了主 DNS 区域会间隔多久来验证辅助区域内的数据有效性，我们也可以手工复制来进行数据同步。

在辅助区域的空白处，单击右键可以看到有两个加载选项，如下图：

那这两个选项有什么不同呢，简单讲，从【主服务器复制】其实只是更新增量的部分，也就是 IXFR 更新，而【主服务器重新加载】则是重新更新所有的记录，也就是 AXFR 更新。测试过程其实很简单，在 DNS1 任意添加一条记录，然后再 DNS2 上使用着脸个选项，就可以看到记录很快被同步过来。具体实验不再演示了，我抓了两张协议分析图以供参考，如下图：

【主服务器复制】：

No.	Time	Source	Destination	Protocol	Info
1	17:14:06	192.168.1.2	192.168.1.1	DNS	Standard query SOA a.com
2	17:14:06	192.168.1.1	192.168.1.2	DNS	Standard query response SOA dns1.a.com
3	17:14:06	192.168.1.2	192.168.1.1	DNS	Standard query IXFR a.com
4	17:14:06	192.168.1.1	192.168.1.2	DNS	Standard query response SOA dns1.a.com

图片看不清楚？请点击这里查看原图（大图）。

【主服务器重新加载】：

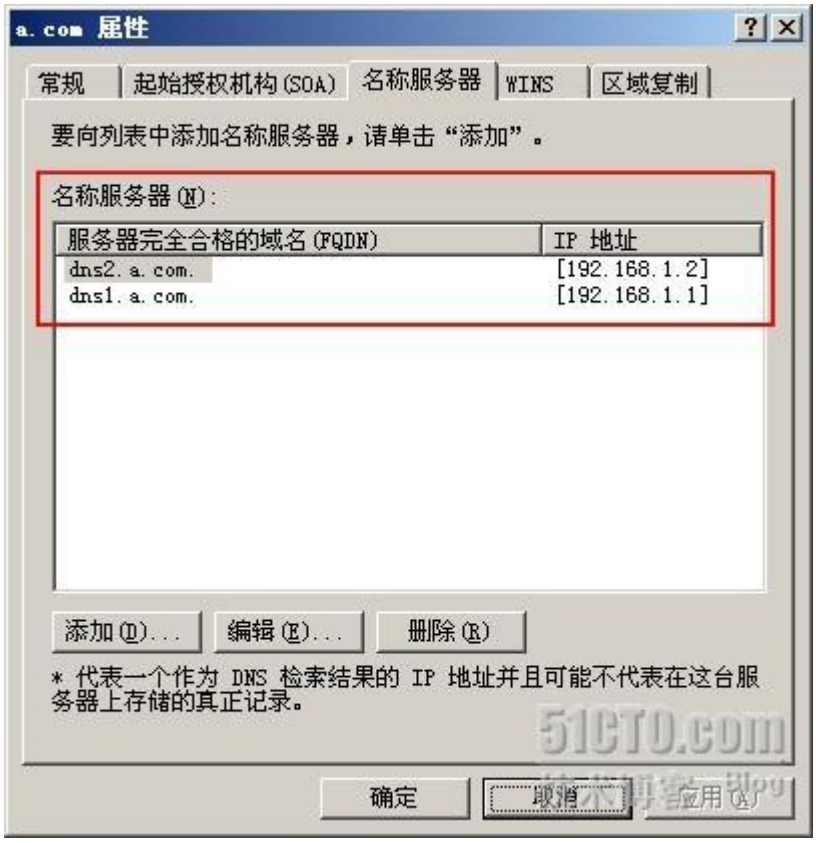
No.	Time	Source	Destination	Protocol	Info
1	17:21:13	AsustekC_37:	Broadcast	ARP	Who has 192.168.1.1? Tell 192.168.1.2
2	17:21:13	Dell_0f:18:7	AsustekC_37:	ARP	192.168.1.1 is at 00:1c:23:0f:18:71
3	17:21:13	192.168.1.2	192.168.1.1	DNS	Standard query SOA a.com
4	17:21:13	192.168.1.1	192.168.1.2	DNS	Standard query response SOA dns1.a.com
5	17:21:13	192.168.1.2	192.168.1.1	TCP	syscomlan > domain [SYN] Seq=0 Win=65535 Len=0 MSS=1460
6	17:21:13	192.168.1.1	192.168.1.2	TCP	domain > syscomlan [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460
7	17:21:13	192.168.1.2	192.168.1.1	TCP	syscomlan > domain [ACK] Seq=1 Ack=1 Win=65535 Len=0
8	17:21:13	192.168.1.2	192.168.1.1	DNS	Standard query AXFR a.com
9	17:21:13	192.168.1.1	192.168.1.2	DNS	Standard query response SOA dns1.a.com NS dns1.a.com A 0.0.0.0 A 1
10	17:21:13	192.168.1.2	192.168.1.1	TCP	syscomlan > domain [FIN, ACK] Seq=28 Ack=187 Win=65349 Len=0
11	17:21:13	192.168.1.1	192.168.1.2	TCP	domain > syscomlan [ACK] Seq=187 Ack=29 Win=65508 Len=0
12	17:21:13	192.168.1.1	192.168.1.2	TCP	domain > syscomlan [FIN, ACK] Seq=187 Ack=29 Win=65508 Len=0
13	17:21:13	192.168.1.2	192.168.1.1	TCP	syscomlan > domain [ACK] Seq=29 Ack=188 Win=65349 Len=0

至此，副本区域已经创建完毕。为了测试方便，我同时也创建了反向查询区域和对应的副本区域。创建完成并重启 DNS 服务，系统会自动将副本的 A 记录同步到区域内，如下图：

图片看不清楚？请点击这里查看原图（大图）。

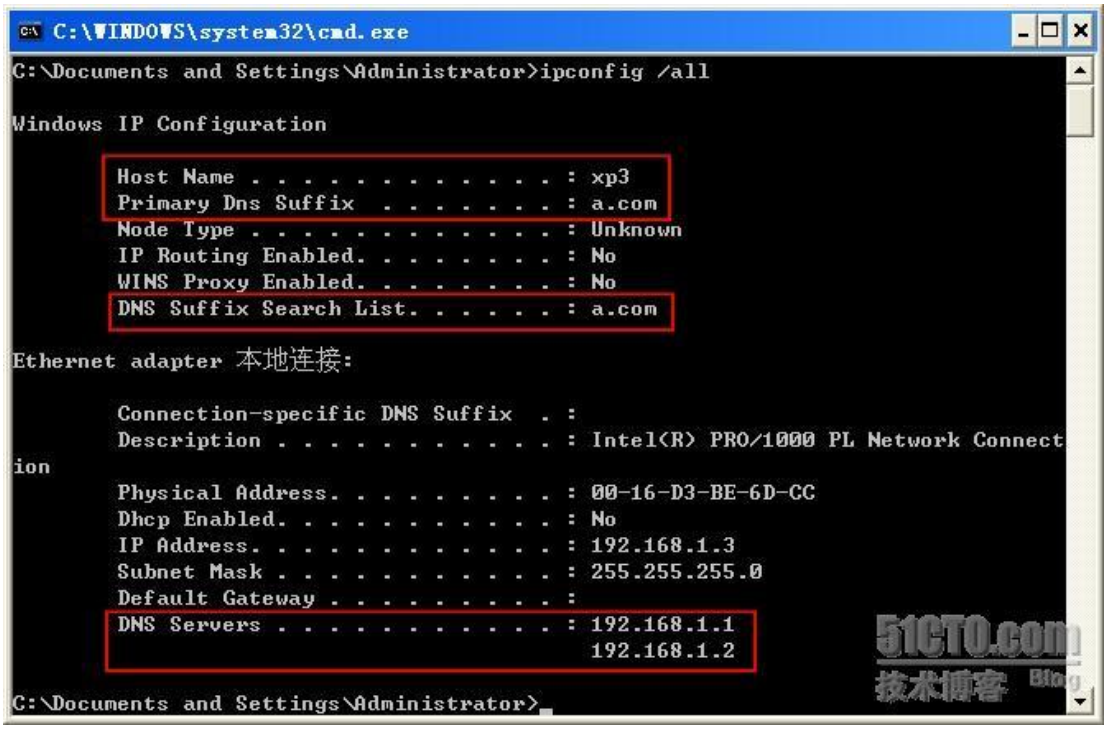
此时，我们可以在主 DNS 区域属性的 NS 选项卡中，将 DNS2 加入进去，毕竟此时它已经是一台解析服务器了，副本区域依然有解析客户端请求的功能。如下图：

正常情况下，可以很快解析到辅助 DNS，确定返回后，如下图：



上图中可以看到，对于 a.com 而言，DNS1 和 DNS2 这两台 NS 服务器都可以响应解析请求，现在我们添加一台 XP3 的客户端，进一步验证此结论。

首先我们需要更改 XP3 主机的主 DNS 后缀（此步骤省略），修改完成并重启，接着就是需要为客户端配置 IP 地址了，具体配置如下图：



配置完成后将 XP3 接入网络，此时刷新 DNS 记录，会发现 XP3 已自动注册到 DNS1 内。如下图：

图片看不清楚？请点击这里查看原图（大图）。

经过短暂的更新，DNS2 中也可以看到这条记录。如下图：

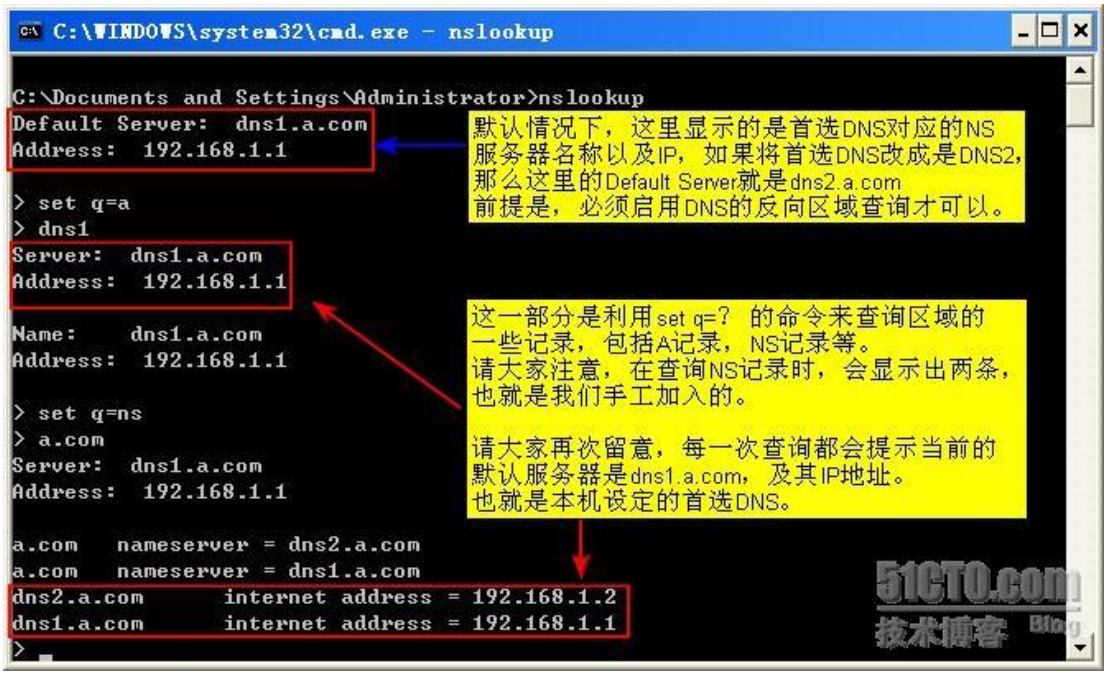
图片看不清楚？请点击这里查看原图（大图）。

那，客户端是如何注册到 DNS 中的呢？还是借助 wireshark 来分析吧，如下图：

图片看不清楚？请点击这里查看原图（大图）。

上图中文字部分大致描述了这个过程。首先是注册到正向区域，其次是反向区域。

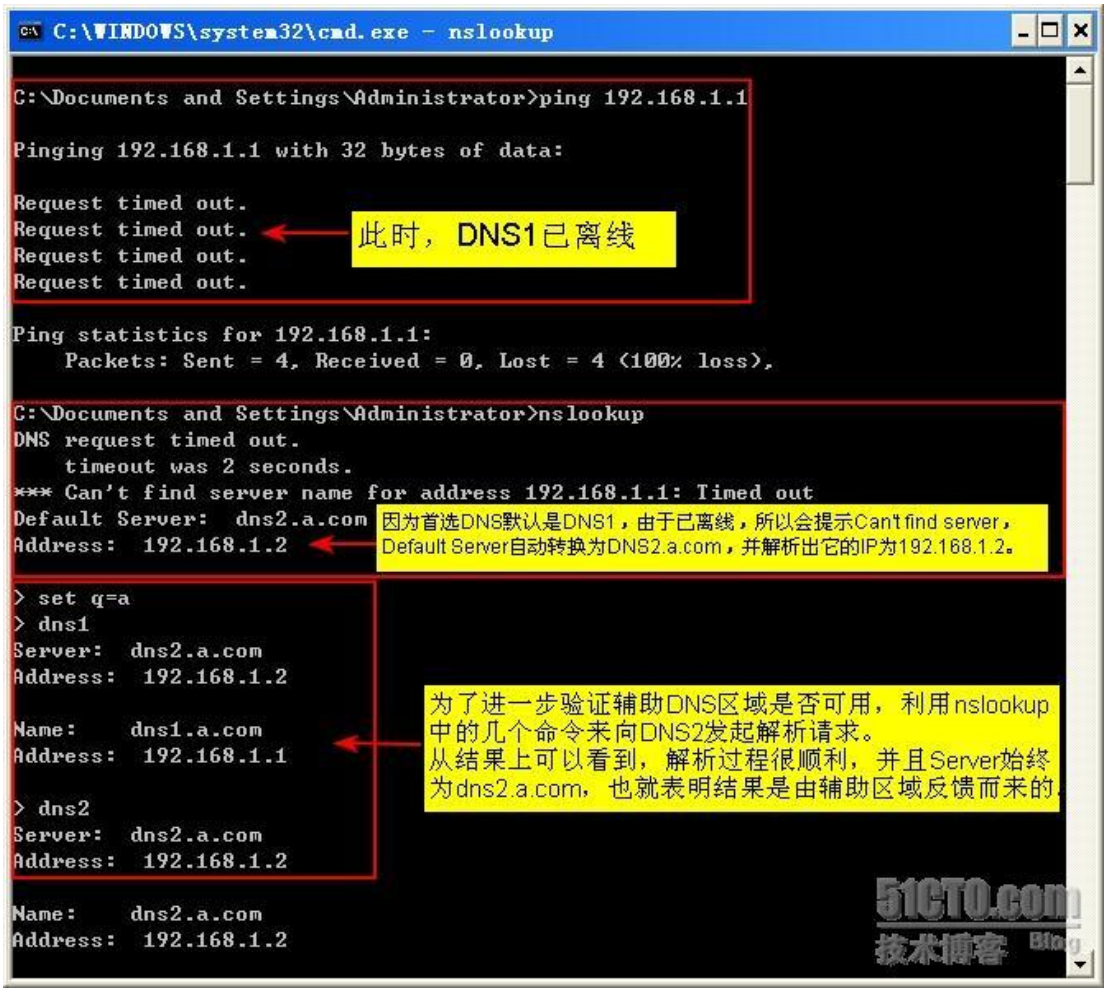
我们在 XP3 上运行一些 DNS 相关的命令，如下图：



图片看不清楚？请点击这里查看原图（大图）。

请大家留意上图的文字部分，内容比较重要。

下面我们要做一个测试，将 DNS1 断开网络，XP3 发出解析请求，看 DNS2 是否可以顺利相应客户端的解析请求。如下图：



图片看不清楚？请点击这里查看原图（大图）。

经过上面的测试表明辅助区域工作正常。

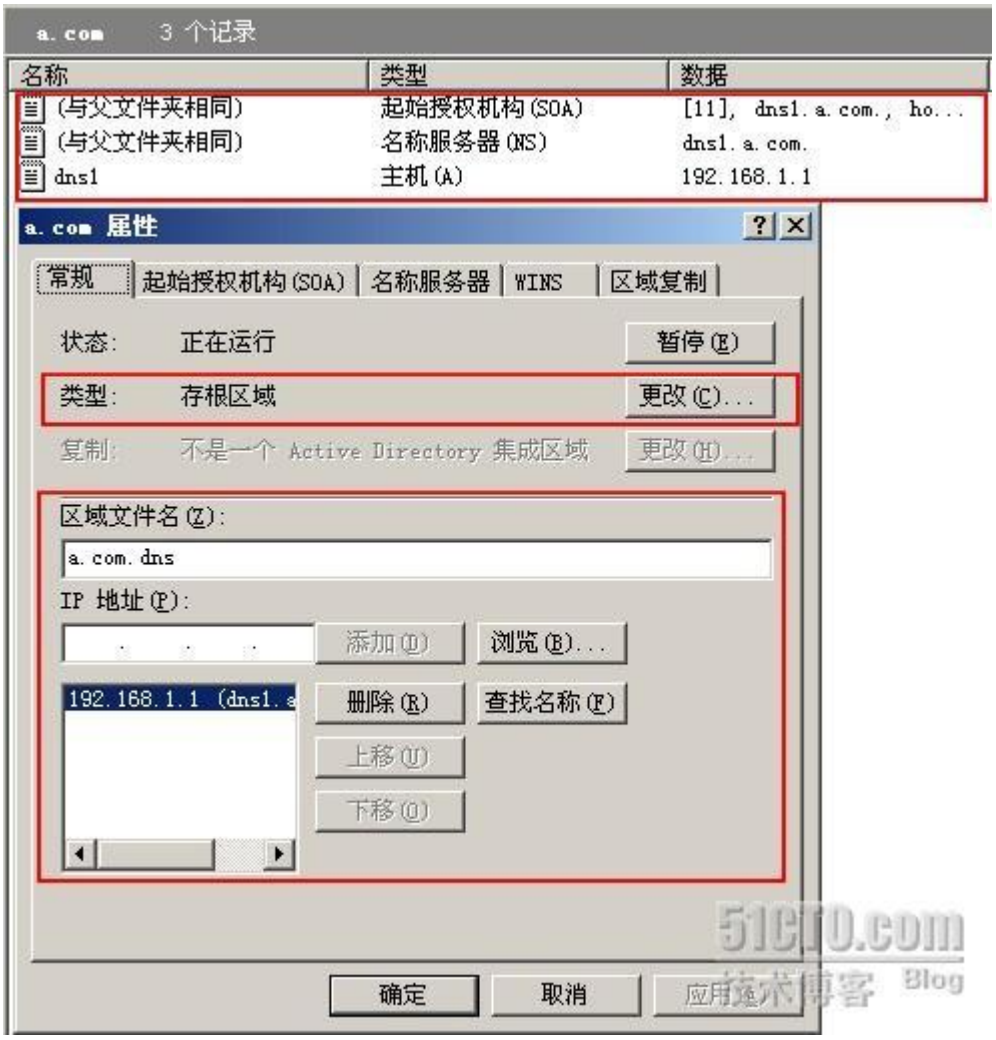
存根区域操作演示

存根区域相对于辅助区域来讲用的不是很多，不过配置也很简单的。存根区域依然是一种副本区域，但与辅助区域不同的是，在存根区域内只保存 SOA、NS 以及 A 记录，也就是说，我们为某一区域的配置存根区域，实际上在生成的存根区域中只包含这三种记录，其他的都不会被复制。这个实验，依然以 DNS1 为主 DNS 服务器，在 DNS2 上创建 a.com 的存根区域。在此之前需要将原有的辅助区域删除。

在区域类型这里我们我们选择【存根区域】，如下图：

这里选择存根区域并点击【下一步】，如下图：

输入区域名称 a.com，几次【下一步】后完成这个向导，打开区域属性。如下图：



区域类型是【存根区域】，记录只有 SOA、NS 以及 DNS1 主机的 A 记录。且各选项卡的值都是灰色，无法更改。如下图：

我们可以在 DNS2 上创建其他区域的存根区域，方法都是一样的。

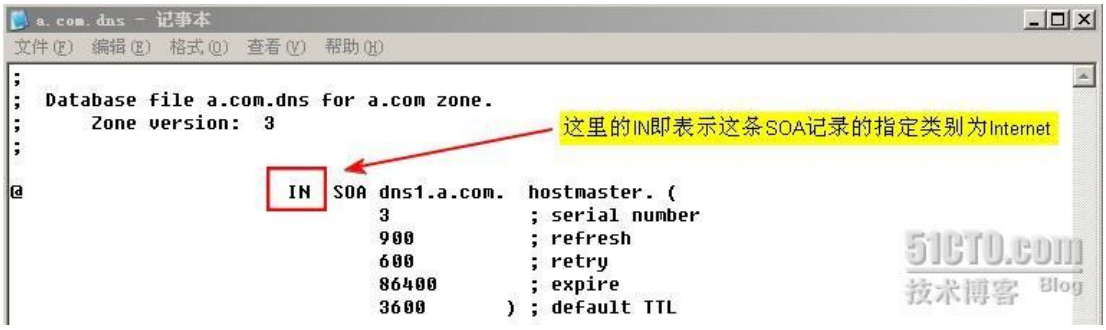
到此，有关辅助区域和存根区域的测试到此为止，希望能帮到大家。

谢谢！

学 DNS 系列（八）DNS 客户端查询过程

DNS 客户端的注册信息在 DNS 服务器中是以记录的方式体现出来的，那么客户端就可以用一些方式进行查询各类记录。相对应的，服务器会对这些查询进行响应，我们称之为解析，至于 DNS 内部的工作机制，我们不得而知，但可以通过一些命令和方法间接地了解 DNS 查询过程。为了更好的描述这个问题，我做了一张简单的 TOPO 图，其中 DNS1 为 DNS 服务器，主机 XP3 是一个 DNS 客户端。拓扑图如下：

在客户端向 DNS 服务器提交一个查询消息中，通常都会包含 3 条基本信息，1、需要制定查询的 DNS 域名；2、指定查询请求的基本类型；3、DNS 域名的指定类别。对于 windows 系统的 DNS 服务器而言，第三条即 DNS 域名的指定类别总是 Internet，即我们常说的 IN 类别，这个关键字无论是在正向还是反向区域的 DNS 文件中都可以看的到，我们以正向查找区域的 DNS 文件为例，如下图：

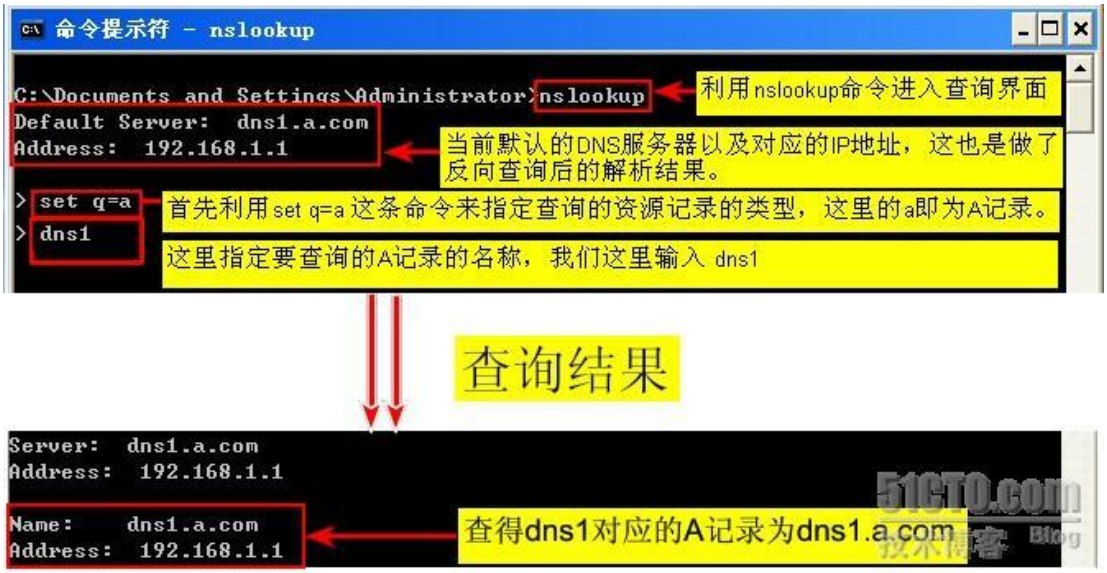


图片看不清楚？请点击这里查看原图（大图）。

所以，通常我们在提交一个查询时只需要包含 DN S 域名和查询请求的类型这两部分内容就可以了，通过一个实例来讲解吧。在 a.com 区域内存在 2 条 A 记录，如下图：

图片看不清楚？请点击这里查看原图（大图）。

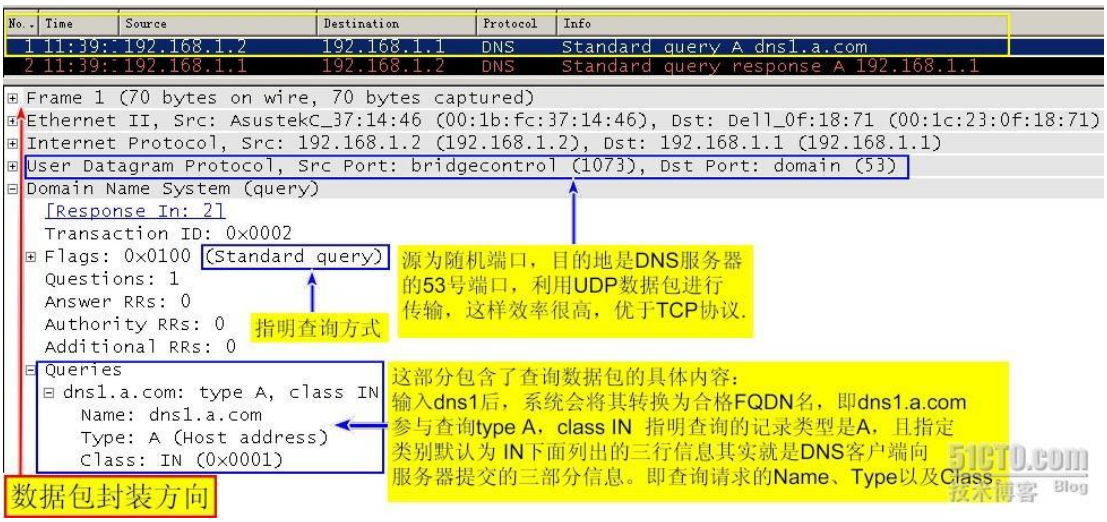
那么我们在 XP3 上查询 DNS1 的 A 记录。为了更好的理解整个过程，我们依然会使用 wireshark 来进行辅助分析。整个过程如下图：



图片看不清楚？请点击这里查看原图（大图）。

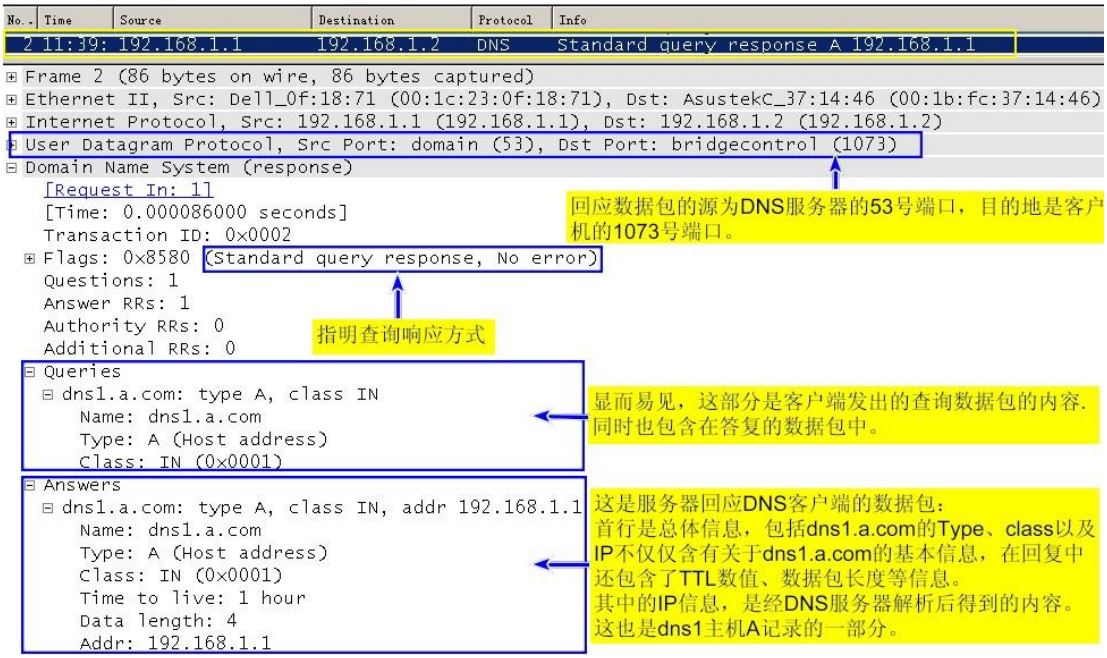
在上图中用到几个命令 ，如 nslookup、set 等，这些命令在 DNS 应用和排错中用的很普遍。所以后面会单独有一节来讲这部分内容。我们再来看看 wireshark 都发生了什么，整个过程经历了 2 个步骤。如下图：

数据包 1 的分析过程：



图片看不清楚？请点击这里查看原图（大图）。

数据包 2 的分析过程：



图片看不清楚？请点击这里查看原图（大图）。

上面是查询 A 记录的过程分析，同理，在查询 NS 记录或者 PTR 指针记录时也是类似的过程，只是查询的类型和结果不同而已，这里不再熬述。

DNS 客户端在第一次查询某条记录时，会按照上述的过程进行，当再次查询时则会首先使用本地缓存信息来解析查询请求，如果有对应的记录则完成此次解析，否则就会向自身的 DNS 服务器发起查询。而这样的本地解析依据来源于 2 个地方，如下：

1、即为本地预先配置的主机名到地址的映射关系，也就是我们常提的 HOSTS 文件。

2、以及系统保存在查询缓存中的记录。

其实，从上面不难看出，客户端解析的过程也就是记录匹配的过程，如果有匹配的选项，则会返回相应信息给客户端，反之则提示无法找到相应记录。

但对于互联网上的 DNS 的查询和解析，就比这个复杂许多，下节会讨论有关迭代和递归查询的内容，敬请期待！

谢谢！

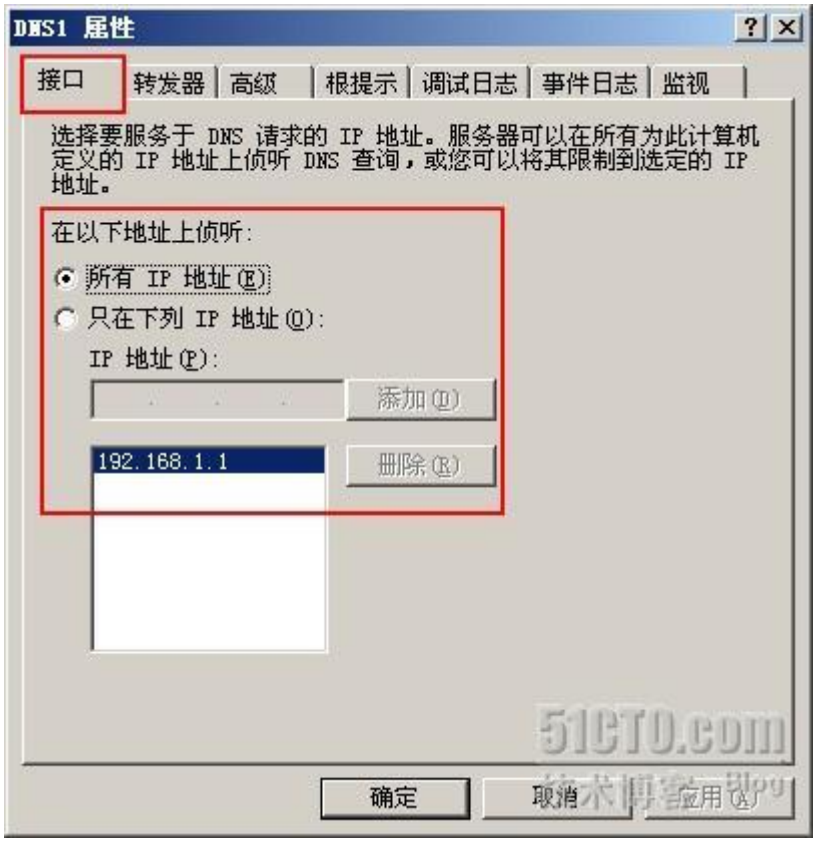
学 DNS 系列（九）DNS 服务器属性之接口和转发器

实验环境未变，TOPO 图如下：

打开 DNS1 的服务器属性选项卡。如下图：

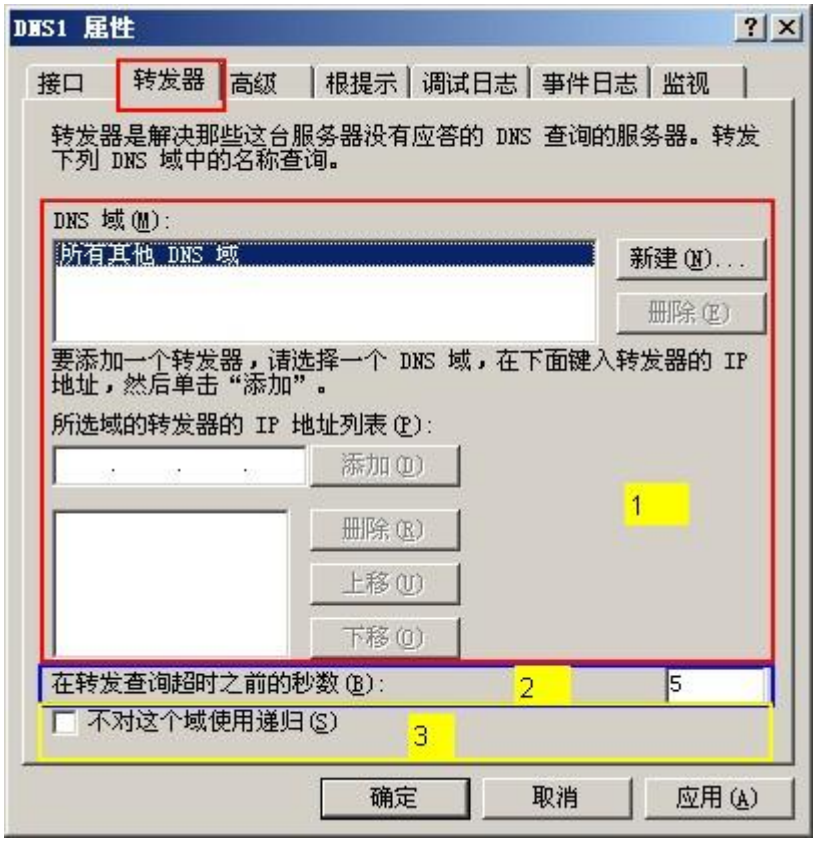

dnsmgmt - [DNS\DNS1]
文件(F) 操作(A) 查看(V) 窗口(W) 帮助(
DNS
DNS1
DNS1
配置 DNS 服务器(C)...
新建区域(Z)...
为所有区域设置老化/清理(S)...
清理过时资源记录(N)
更新服务器数据文件(U)
清除缓存(E)
启动 nslookup (A)
所有任务(K)
查看(V)
从这里创建窗口(W)
删除(D)
刷新(F)
导出列表(L)...
属性(R)
帮助(H)

选定【属性】后，如下图：



在首个选项卡中，我们可以设置那些 IP 地址可以接受 DNS 查询和解析请求，默认情况下是网卡上所有 IP 地址均可，但我们也可以手动指定部分地址。

下一个是【转发器】选项卡，相比前一个要略显复杂。如下图：

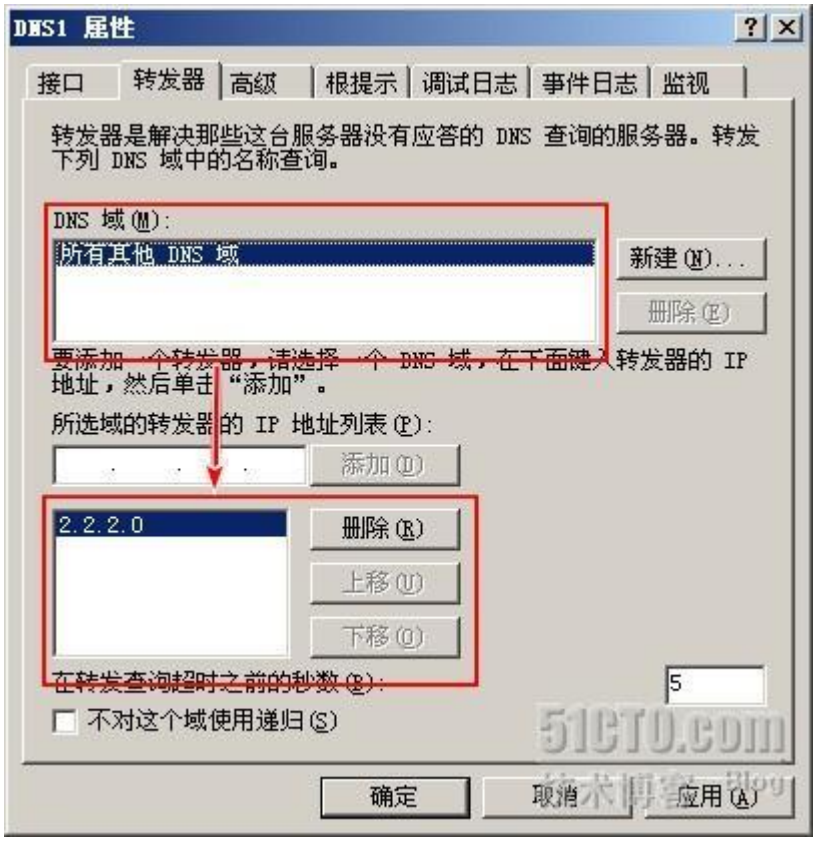


我用 3 种颜色的方框将这个选项卡的分成了三个部分，转发器的目的是，当此 DNS 遇到非本 DNS 负责的区域的查询请求时，服务器会将解析请求信息发到转发器上，同时需要检查两个方面。

1、是否满足转发器设置的区域范围要求，默认情况下，DNS 会接受除自身区域外的其他 DNS 域的请求，如红色框图的部分。我们也可以指定只对某一个或某些 DNS 区域进行转发，可以通过点击上图的【新建】选项来添加区域名称。一般情况下，这里都是默认全部区域的。2、同时会检查设定区域对应的转发 IP 地址，也就是下一个 DNS 服务器的地址，默认情况是下一个 DNS 服务器的地址为空。

此时，DNS 并非就此罢工，因为当我们输入一个域名后，DNS 服务器对此域名采取的解析方式是多种多样的，比如 NBNS 解析等。如果此处没有设置转发地址，则会使用配置中的 13 个根提示服务器尝试进行解析，但请大家注意，此时需要为 DNS 服务器设置一个网关或路由器地址，这样才能将信息交由外部 DNS 代为处理。

我们来模拟一下这个过程，现在将 DNS 和客户机的网关都设置为 192.168.1.1，并在此基础上为 DNS 配置一个转发地址 2.2.2.0，在客户端，我们通过浏览器访问 op.com，我虚拟域名。首先，设置添加一个转发地址，如下图：

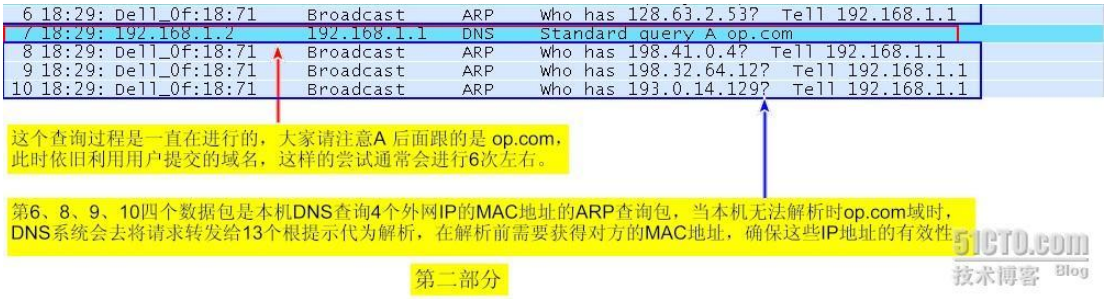


然后，我们在 XP3 上利用浏览器访问 op.com，并用 wireshark 来观察数据包的变化，因为内容很多，所以我将这个关键过程分为四部分，分别说明。如下图：

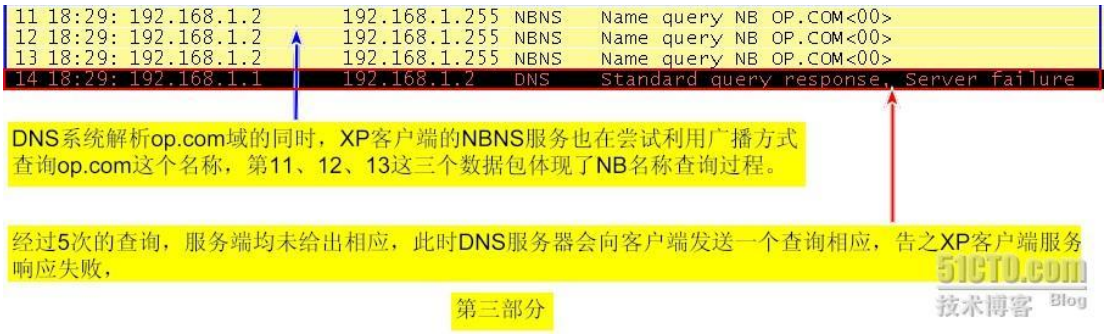
第一部分，如下图：

图片看不清楚？请点击这里查看原图（大图）。

第二部分，如下图：



第三部分，如下图：



图片看不清楚？请点击这里查看原图（大图）。

第四部分，如下图：

图片看不清楚？请点击这里查看原图（大图）。

除了添加.com 的顶级域外，还会有如.net、.org 等，之后还产生了很多数据包，但基本大同小异，所以不再详述了。从这几个图可以得知，客户端发起一个域名解析请求，除了 DNS 忙于解析和响应外，同时系统也在试图寻求别的解析方式，目的都是利用现有的系统配置充分地完成客户端的解析请求。

在【转发器】选项卡的下方，有一个设置超时的选项，默认是 5 秒，也就是说自从尝试第一个转发器 IP 开始，在这个时限内如果没有解析成功，则会自动转向第二个 IP 地址进行解析，以此类推。比如现在另加一个 IP3.3.3.0，同时将这个时间设为 2 秒。如下图：

图片看不清楚？请点击这里查看原图（大图）。

此选项卡的最后一个属性是询问是否对此域使用递归查询，也就是说如果转发器发生故障，则 DNS 服务器不会代替客户端进行下一步的查询，也就是寻找下一级 DNS 服务器进行解析，客户端则一直处于等待状态，直到 DNS 服务器把正确的解析结果反馈给客户端为

止，整个查询过程才算结束。为了保证查询的准确性，默认是允许递归的，但当客户端数量庞大，解析请求非常频繁时，DNS 服务器会在资源消耗上会有不小的压力，请大家留意这里。

这里提到了一点有关递归查询的内容，还有迭代查询以及 DNS 循环等内容会在下面的部分讲到，敬请期待！

说明：本文中举例的转发器的 IP 地址 2.2.2.0，这个是网络地址无法作为 IP 地址使用，在真实环境中请大家使用正确的 IP 地址，此处仅为举例，特此说明。

谢谢！

学 DNS 系列（十）图、例详解 DNS 递归和迭代查询原理及过程 (1)

上节中提到了一些有关递归查询的内容，但说的很少，也很笼统，本节将会从原理和实例两方面入手分析 DNS 的递归以及迭代查询。

在此之前，我们需要了解一些背景知识，以便于更好的理解今天的主题内容。

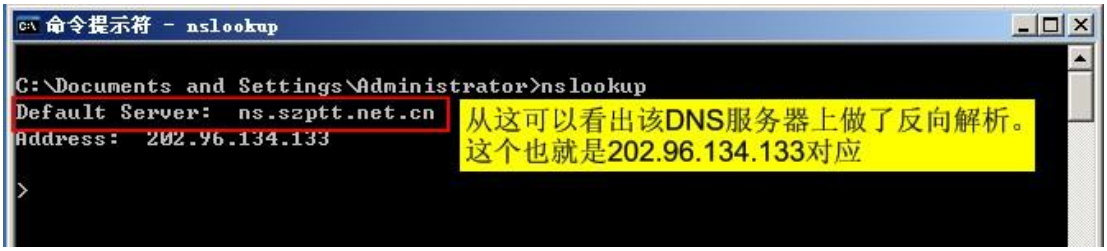
在互联网中，一个域名的顺利解析离不开两类域名服务器，只有由这两类域名服务器可以提供“权威性”的域名解析。

第一类就是国际域名管理机构，也就 InterNIC，主要负责国际域名的注册和解析，第二类就是国内域名注册管理机构，在中国就是 CNNIC 了，主要负责国内域名注册和解析，当然，尽管分为国际和国内，但两者一主一辅，相互同步信息，毕竟最终的目的是在全球任何一个有网络的地方都可以顺利访问任何一个有效合法的域名，其间的联系就可见一斑了。

有的朋友可能会有这个疑问，域名服务器不是有很多吗？为什么说只有 2 类呢？是的，ISP 何其多？当我们输入某一网址（或域名），系统将这个域名发送至需要将其当前已配置的 DNS 服务器，以便转换为 IP 地址进行访问，通常会是当地的公共 DNS 服务器（内网环境可能直接提交到防火墙或路由器上做进一步转发处理）。公网 DNS 服务器收到此请求后，并非立刻处理，比如转发至上一级的 DNS 服务器（在第一节讲过 DNS 有着很严格的逻辑层次关系），而是首先会查看自己的 DNS 缓存，如果有这个域名对应的 IP，则直接返回给用户，系统收到这个 IP 后交给浏览器做进一步处理。在这个轮回的过程中，客户端所得到的 DNS 的回复就是“非权威的性”的，也就是说这个结果并不是来自这个域名所直接授权的 DNS 服务器，而是该记录的副本。简单的说，“非权威性”的应答是从别的 DNS 服务器上复制过来的，与之对应的，就是“权威性”应答则是由域名所在的服务器作出的应答，听起来似乎不易理解，我们来看一个例子。

我所在地是深圳，这里的公共 DNS 服务器是 202.96.134.133，我们来检测一下。

如下图：

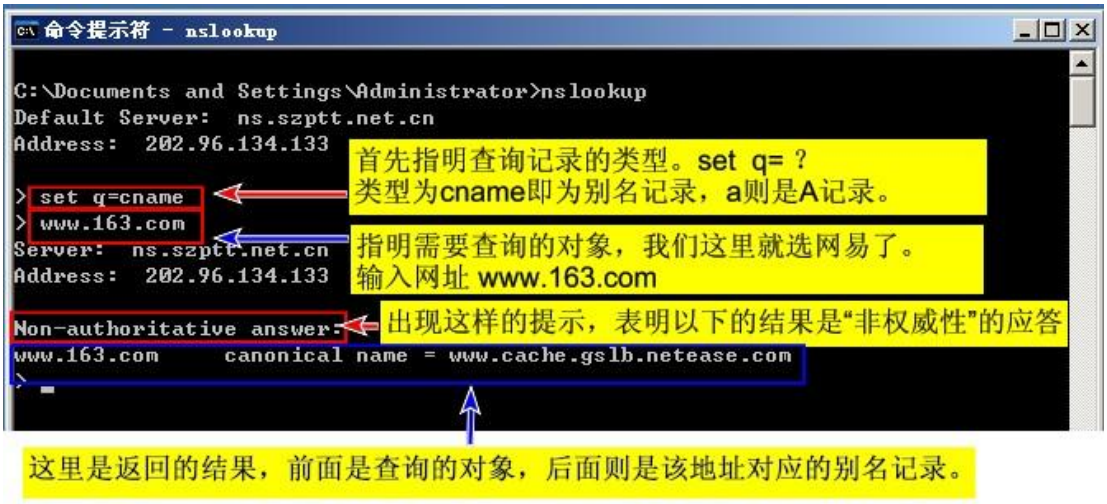


图片看不清楚？请点击这里查看原图（大图）。

这里用到了 nslookup 命令，用来查询当前本机解析域名所依赖的 DNS 服务器，从上图中文名可以得知当前默认的 DNS 解析服务器是 ns.szptt.net.cn，对应的 IP 地址为 202.96.134.133，也就是说在这台机子上运行的网络程序，如果需要用到 DNS 域名解析的，都会将请求到这个服务器上，寻求解析。

当然，如果你是在内网，或是其他类型的局域网，在解析时候可能无法顺利得到上图的结果，多半是代理或防火墙的缘故。建议 ADSL 用户可以自测一下，加深印象。现在，我们来解析一个网站的别名记录，以此来了解一下何为“非授权记录”

以网易为例吧。如下图：



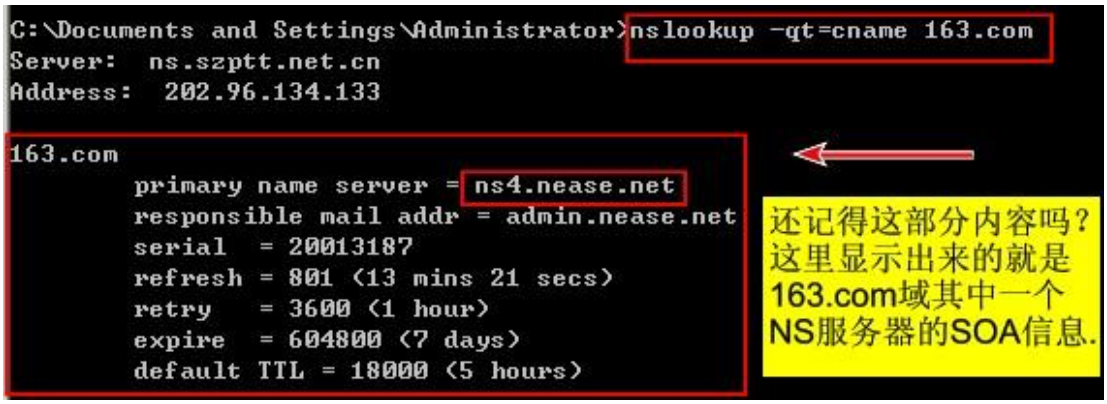
图片看不清楚？请点击这里查看原图（大图）。

依照上图步骤就完成了一次 CNAME 记录的查询，通过这次小测试，希望大家注意一下几点：

1、查询的命令不仅这一种，我们还可以用命令 nslookup -qt=cname www.163.com ，返回的结果是一样的。

2、查询的对象需要是一个完整的 URL 地址，而并非域名，如果想查询对象写出 163.com，则默认值返回 163.com 这个域的 ns 记录。

如下图：



图片看不清楚？请点击这里查看原图（大图）。

163.com 的子域名还有很多，不同的子域名可能对应不用的 NS 服务器，这样做的目的是可以更快的相应客户请求，这就用到的服务器的均衡负载技术了。所以网易的 NS 服务器也肯定不只这一个。可以用命令来证实，如下图：

图片看不清楚？请点击这里查看原图（大图）。

从上图可知，网易的 NS 服务器至少有 2 台。

以上所有的信息都是“非权威性”的回应，换句话说，这些记录都保存在深圳的这台 DNS 服务器上，刚才查询的所有结果均来源于此，自然都是副本信息。

那如何才能找到最原始的解析记录呢？要想揭开这个疑难，我们需要对 DNS 的查询原理有一定的认识。下面是是 DNS 查询的大致步骤：

1> 首先，客户端提出域名解析请求（无论以何种形式或方法），并将该请求发或转发给本地的 DNS 服务器。

2> 接着，本地 DNS 服务器收到请求后就去查询自己的缓存，如果有该条记录，则会将查询的结果返回给客户端。（也就是我们看到的““非权威性”的应答”）。

请注意，下面就开始递归查询了：

反之，如果 DNS 服务器本地没有搜索到相应的记录，则会把请求转发到根 DNS（13 台根 DNS 服务器的 IP 信息默认均存储在 DNS 服务器中，当需要时就会去有选择性的连接）。

3> 然后，根 DNS 服务器收到请求后会判断这个域名是谁来授权管理，并会返回一个负责该域名子域的 DNS 服务器地址。比如，查询 ent.163.com 的 IP，根 DNS 服务器就会在负责.com 顶级域名的 DNS 服务器中选一个（并非随机，而是根据空间、地址、管辖区域等条件进行筛选），返回给本地 DNS 服务器。可以说根域对顶级域名有绝对管理权，自然也知道他们的全部信息，因为在 DNS 系统中，上一级对下一级有管理权限，毫无疑问，根 DNS 是最高一级了。

4> 本地 DNS 服务器收到这个地址后，就开始联系对方并将此请求发给他。负责.com 域名的某台服务器收到此请求后，如果自己无法解析，就会返回一个管理.com 的下一级的 DNS 服务器地址给本地 DNS 服务器，也就是负责管理 163.com 的 DNS。

5> 当本地 DNS 服务器收到这个地址后，就会重复上面的动作，继续往下联系。

6> 不断重复这样的轮回过程，直到有一台 DNS 服务器可以顺利解析出这个地址为止。在这个过程中，客户端一直处理等待状态，他不需要做任何事，也做不了什么。

7> 直到本地 DNS 服务器获得 IP 时，才会把这个 IP 返回给客户端，到此在本地的 DNS 服务器取得 IP 地址后，递归查询就算完成了。本地 DNS 服务器同时会将这条记录写入自己的缓存，以备后用。

到此，整个解析过程完成。

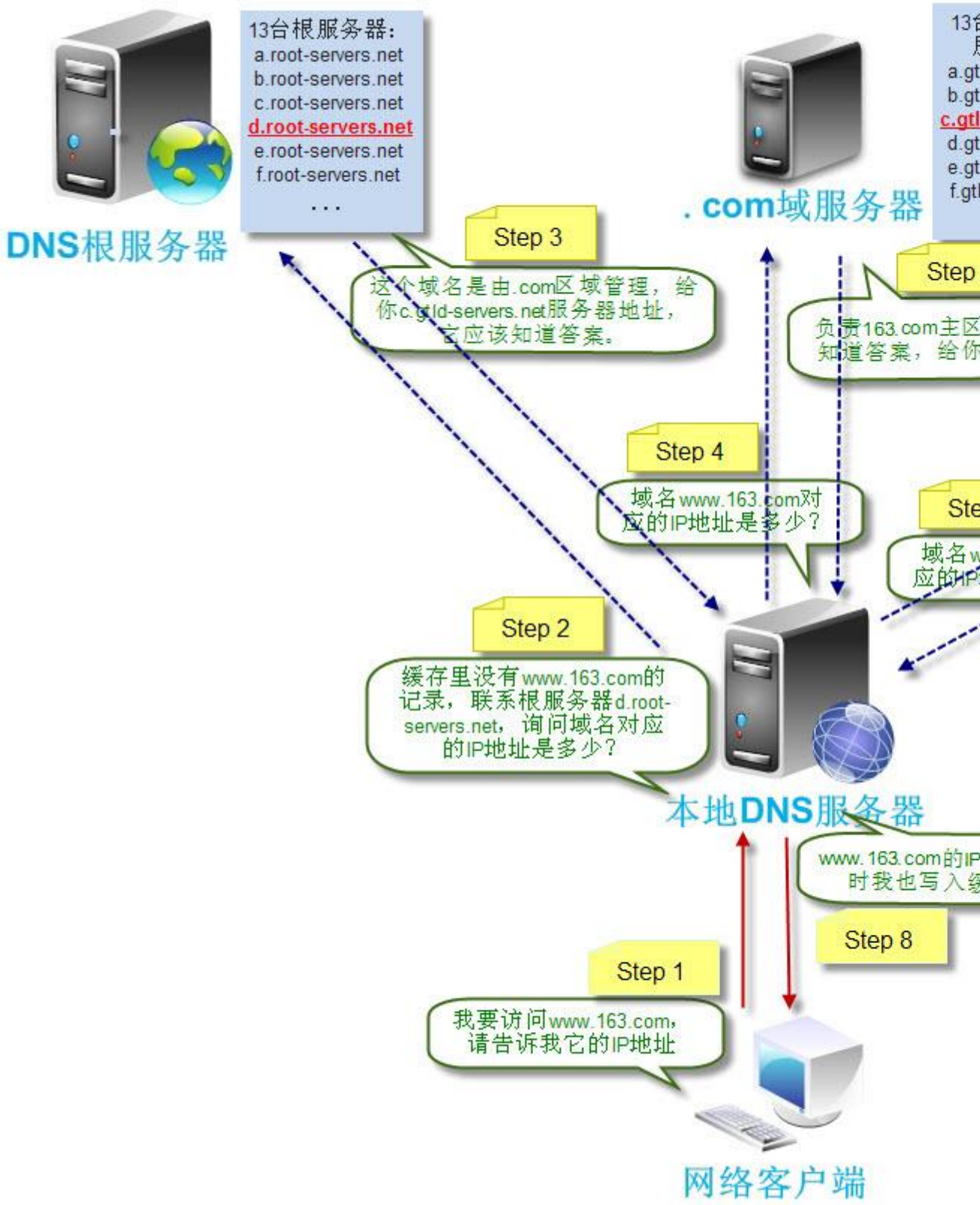
客户端拿到这个地址后，就可以顺利往下进行了。但假设客户端请求的域名根本不存在，解析自然不成功，DNS 服务器会返回此域名不可达，在客户端的体现就是网页无法浏览或网络程序无法连接等等。

下节中，我将以图解的方式将这个过程体现出来，便于大家形象化的理解递归查询的过程，谢谢。

敬请期待！

一起学 DNS 系列（十）图、例详解 DNS 递归和迭代查询原理及过程 (2)

上节里，我们用文字的形式大致描述了 DNS 解析的过程，为了更好的帮助大家理解整个解析过程，我做了一张 DNS 域名解析的分步图，如下：

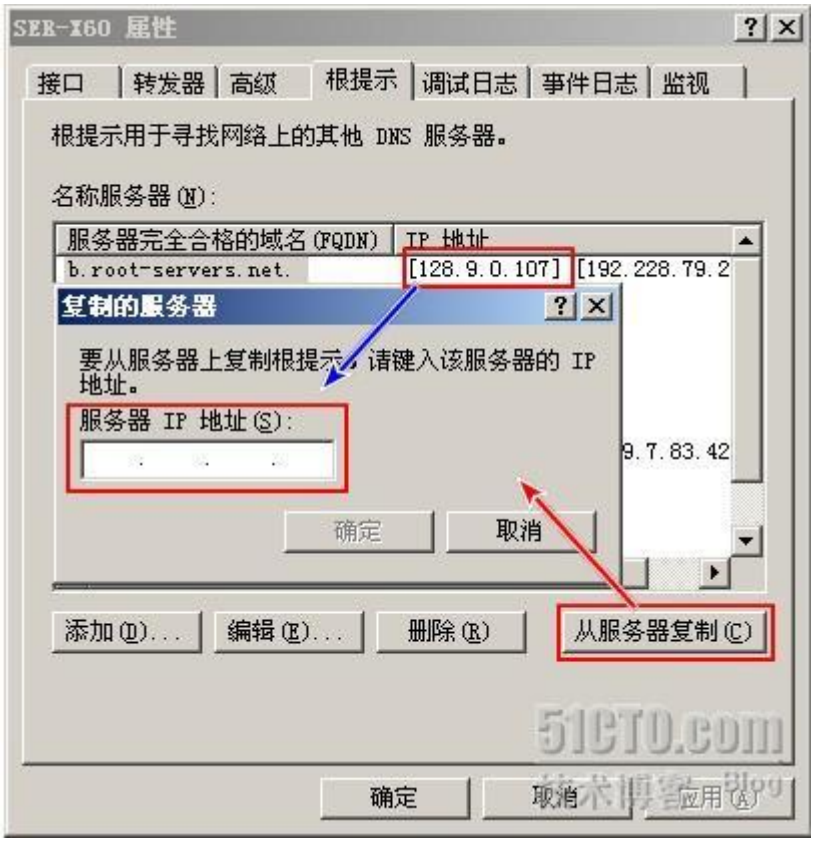


在这个图里，通过 8 个步骤的解析过程就使得客户端可以顺利访问 www.163.com 这个域名，但实际应用中，通常这个过程是非常迅速的，主要由几个方面的原因所决定。1、客户端网络状况是否良好，2、与本地 DNS 连接的速度是否优秀，3、本地 DNS 上是否有访问地址的缓存等

等，如果以上的因素答案都是肯定的，那么访问就会很迅速，上图的步骤也会骤减至 2 个，因为有缓存，所以本地 DNS 服务器会很快告之域名对应的 IP 而实现迅速访问。

上图中出现了 2 个陌生的列表，下面就说说这两张表的来历。这里我们结合第九章的内容继续讲解 DNS 的高级属性，如下图：

可以看到，在【根提示】选项卡中列出了 13 台根服务器，分别是 (a~m).root-servers.net 和对应的 IP 地址，有的是 2 个 IP，后面那个是备选地址，我们可以手动修改这些地址，但一般情况下，建议不要去动它。如果不小心更改或者删除，我们还是有几个办法修复的。因为这些服务器的地址列表是整个互联网共享的，所以我们可以在网络上找到最新的根服务器列表。通常在这个链接里：ftp://rs.internic.net/domain/named.root ，也可以通过直接从网络上复制。如下图：



在服务器 IP 地址里，我们可以输入 13 个地址中的任意一个，确定后系统会自动连接到该服务器上更新列表。也并非 13 个地址中的一个，如果同网段内有冗余 DNS，这里就可以输入那台 DNS 的地址，也是可以更新的。前提是，两台 DNS 服务器都必须连接到互联网。当然在 DNS 的安装目录下的 CACHE.DNS 文件中也是可以找到的，具体路径如下：C:\WINDOWS\system32\dns\CACHE.DNS。以上的方法都可以恢复这个列表。

在回到第一个图中，当本地 DNS 服务器向根 DNS 查询时，它会搜索自己的根 DNS 服务器列表，找到一个连接的地址，比如 d.root-servers.net，这样就联系到了根服务器，当然，连接其他的也可以，没有太大区别。根服务器检测到是.com 域名后，就返回给本地 DNS 服务器一个 IP 地址，这个 IP 就是负责.com 顶级域名的其中一个服务器，我这里选的是 c.gtld-servers.net，同样的，一共有 13 台这样的服务器负责.com 域名的解析，即(a~m).gtld-servers.net。可能有的朋友疑问，这个是这么知道？OK，要解这部分内容，我们需要用到另一个工具 dig，这个原本是 Linux 下的 DNS 服务器的调试工具，类似 windows 下的 nslookup，但功能上比后者强很多，我们先做个演示，至于如何使用，后面会有章节来描述。

我们用 dig 命令来跟踪一下到 www.163.com 网站的整个过程，如下图：

图中提到的 gTLD，其实这是顶级域名的一个分类，除此之外还有 ccTLD，也就是国家及地区代码顶级域名，即 CountryCodeTLD，比如.cn 表示中国.hk，表示香港等。上图的 4 个过程其实就是我们从提交请求，到正常访问的过程。上图中还有很多参数没有说明，这部分会再后面章节有详述。

现在我们再来说一下递归查询和迭代查询。

在本节的第一张图中，当本地的 DNS 服务器帮助客户端解析 www.163.com 这个地址的 IP 地址的过程中，其实有已经包含了这 2 类查询。从客户端到本地 DNS 服务器是属于递归查询。而 DNS 服务器之间就是的交互查询就是迭代查询。

我们模拟一个场景。比如你的老板要去喜来登大酒店，但不知道怎么走，于是问你（你是他的秘书），此时你也不知道，于是问张三，张三也不清楚，让说让你去问李四，于是你问李四，李四正好知道，然后把线路告诉你，然后你把结果告诉你老板，这样整个问询就完成了。那么这就是个递归的过程。在这个过程中，老板始终在等待你的答案，而自己却丝毫不关心这档子事，而你就充当了一个代理和中间人的角色，来全权负责此时，你的目的是要把答案找到并反馈给老板。

你、张三、李四这三个人之间的信息传递，就是迭代的过程，因为你在问张三的时候，张三并没有像你代替你老板一样去问别人，而只是返回给你一个参考答案。这样的问询方式，我们就称之为迭代查询。

在默认情况下 DNS 服务器可以接受来自其他客户机（或其他 DNS 服务器）的迭代或递归查询，如果流量较大的服务器通常都只接受迭代查询，比如 13 台根服务器。因为如果它们对每一个解析请求都代为查询的话，那将会消耗极大的服务器资源，可能会导致服务器过载甚至崩溃。

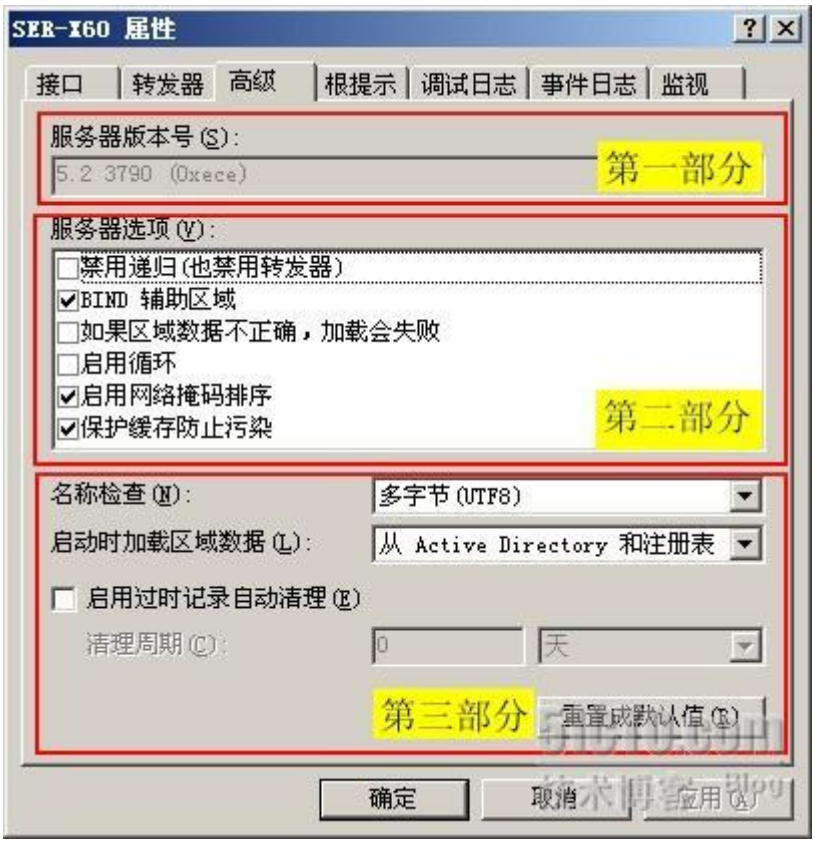
关于 DNS 的迭代和递归查询我们先聊到这里。下节继续讨论 DNS 服务器高级属性的其他选项卡，敬请期待！

本文内容比较集中，如果有未讲到的地方，请大家指出，我会及时补充，谢谢！

一起学 DNS 系列（十一）DNS 服务器属性之高级服务器选项

接着第九节的未完的部分，我们来继续探讨 DNS 服务器的属性，今天主要内容是【高级】选项卡，也是 DNS 属性比较重要的一部分，但有趣的是，我们平时却很少修改这里的内容。不过，各个选项的含义还是需要认真理解的。

打开 DNS 属性，找到【高级】选项卡，如下图：



这个是【高级】选项卡里的内容，这里我们一一进行讲解。

第一部分：

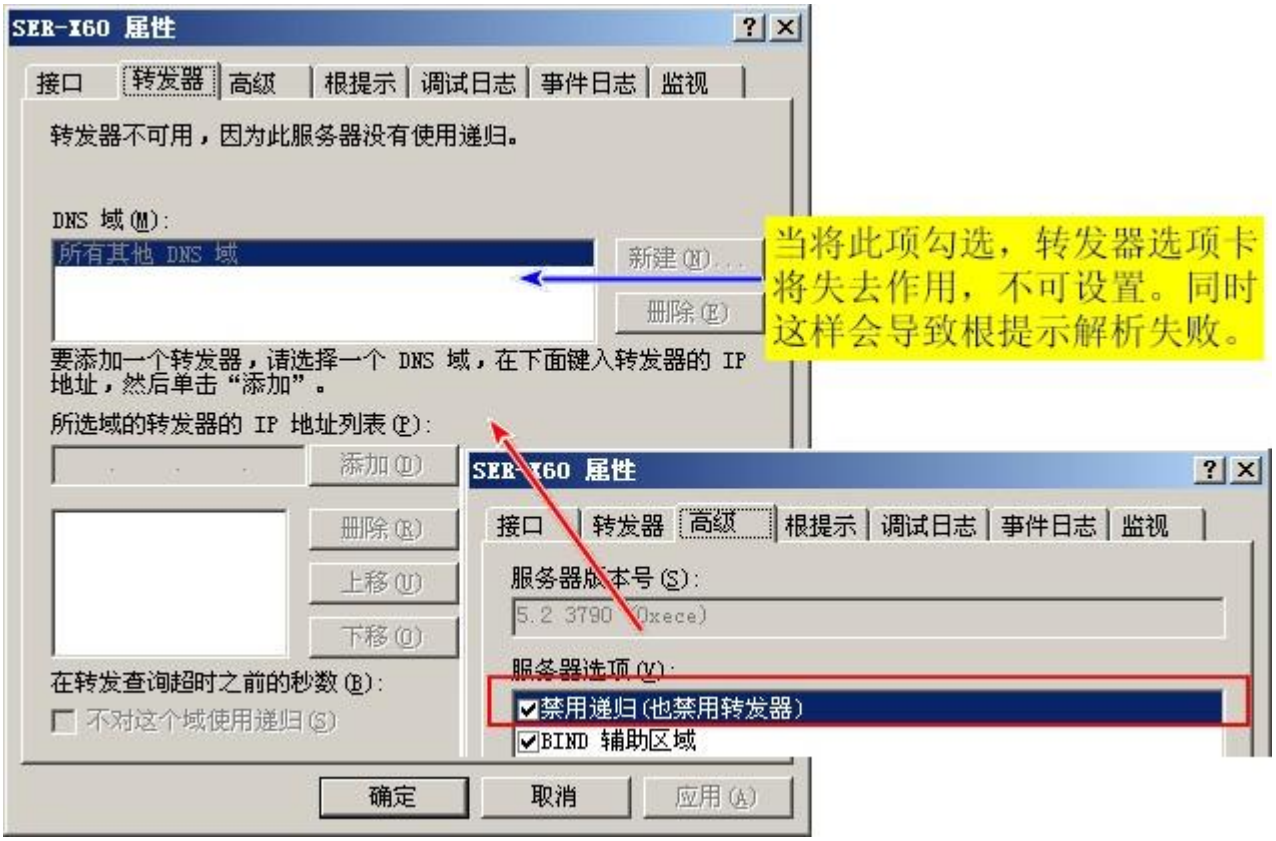
这里的服务器版本号主要是为了电话咨询、疑难解答时候，告知工程师这个版本号，便于解决问题。其实也就是在寻求 MS 技术支持服务时会用得到的。在版本号 5.2.3790（0xece）中，5.2 是整个系统的版本号，后面的 0xece 就是十六进制表示方式，换成十进制就是 3790。这部分东西很少，了解即可。

第二部分：

看设置项目的多少就知道这部分是重点啦，的确如此。在这里我们来对每一个选项进行描述，重点地方重点讲解。（请注意，以下所有设置均不保存。）

☐禁用递归（也禁用转发器）

上两节里，叙述了有关 DNS 的递归和迭代查询的内容，这个选项也与之有关。如果勾选，那么也就是说关闭了 DNS 服务器的递归，同时【转发器】选项卡也将失去作用，这样 DNS 服务器就只能工作在迭代查询的模式下了。也就是只能相应查询请求，而不代为查询，默认为启用此项。如下图：



☐BIND 辅助区域

BIND 是 linux 下 DNS 服务器，我们可以讲两个系统的 NS 服务器联合起来使用，但由于 Windows 下的 DNS 在在区域传输时使用快速传输格式，这与 BIND 有一定差别，不过 BIIND 4.94 以上的版本都支持这种格式，所以这是个保留选项，是为了兼容早版本的 BIND 服务器。

☐如果区域数据不正确，记载会失败

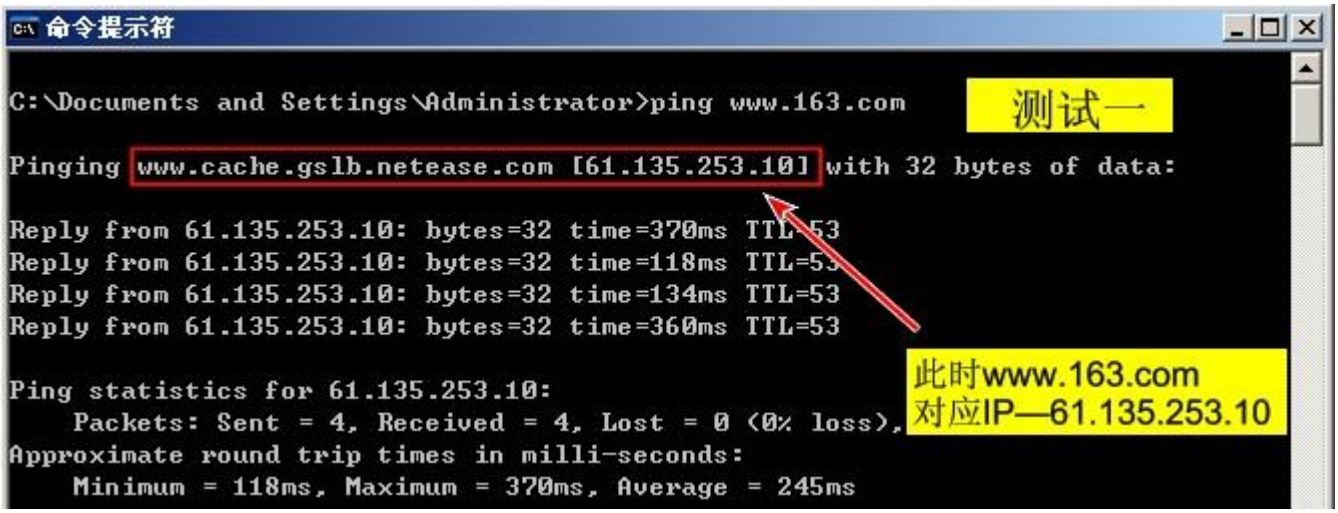
这一项比较好理解，当选中此项后，在 DNS 在加载区域数据时，如果数据有问题，则会停驶加载，并出现错误提示，反之则继续之前的操作，即使数据不正确。显然，这并不是我们想要的结果，所以默认情况下此项未启用，目的是保证 区域数据的正确性，同时也是为了可以给客户端提供正确的请求结果。

☐启用循环

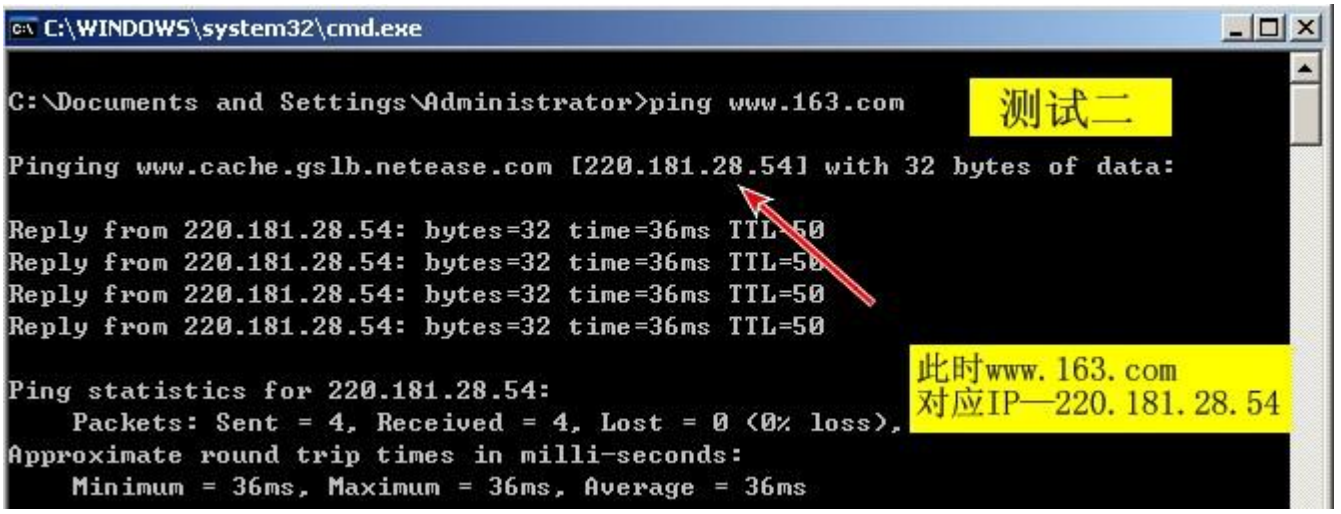
默认情况下，这项设置是启用的，如果你的服务器中有多条类似的记录且访问量很大的话，建议使用此功能。英文版的系统中被称为 Round Robin，提到这个可能不少朋友都有接触或了解，那具体是做什么的呢？

不知道大家是否有这样的经历，在检测网络通断的时候，常用的方法是用 ping 一个网站，比如网易，细心的朋友可能会发现，每次 ping 所返回的 IP 地址不一样。

这里来做一个测试，如下图：



从上图可以看到 www.163.com 的 IP，然后我换了一台机器，再次运行 ping，如下图：



大家可以看到，此时返回的 IP 已经不再是先前那个了，这是为什么呢？同样的一个网站，地址一样，但 ping 的返回值却不一致。那，原因何在？

其实，如果在 DNS 中启用了循环功能，就可以实现这样的效果。对应 163.com 这个域名而言，在创建 www 主机的时候，就同时对应了多个 IP 地址，比如 1.1.1.1、2.2.2.2、3.3.3.3，在 DNS 的配置文件中体现如下：

```
www     IN    A     1.1.1.1
                    2.2.2.2
                    3.3.3.3
```

当第一个客户请求 www 的解析时，按照默认顺序进行相应，也就是系统会自动返回 1.1.1.1 给客户端，当第二个客户请求 www 的解析时，由于启用了 DNS 循环，所以 DNS 会将这个列表做循环处理，如下所示：

```
www     IN    A     2.2.2.2
                    1.1.1.1
                    3.3.3.3
```

这样一来，返回的 IP 就是 2.2.2.2，同理，第三个客户端得到的 IP 可能就是 3.3.3.3。

从上面例子可以看到，DNS 循环功能其实算是一种很简单的负载平衡处理机制，通过这个方法可以起到分担服务器压力的作用。

当然，163 所采用负载技术远比这个复杂，生产环境中对 DNS 性能负载的处理的方法也多种多样，这里仅仅是个范例而已。

□启用网络掩码排序

从这一设置是 win2003 新增的一个功能，简单的说，针对客户端请求的解析，当存在多个匹配记录时，系统会自动这些记录与客户端 IP 的网络掩码匹配度，按照最相似的原则，来应答客户端的解析请求。来举个例子，这里有某个域名的 A 记录解析表：

```
www    IN    A    161.23.222.51
                  220.22.120.33
                  61.135.251.15
```

当一个用户试图解析这个 A 记录时，如果他的 IP 是 220.22.120.149，系统收到这个解析请求，就会把这个 IP 和列表中的记录进行掩码接近度匹配，他很快发现第二条记录和这个客户 IP 很接近，于是就会把 220.20.120.33 作为结果返回给客户端。这就是 DNS 的网络掩码自动排序功能。此功能是对来访者实行的本地子网优先级匹配原则，这样将最接近客户端 IP 的记录返回给对方，旨在加快客户端的访问速度和效率。

同样，此项设置默认也是自动开启的。

这里还有一点要说明，当同时启用了循环和网络掩码排序，掩码排序优先级高于循环，此时循环则仅作为一种后备方式而存在，如果来访者 IP 都无法匹配，则会采用循环的方式进行答复来访者。

□保护缓存防止污染

听起来有些抽象，解释一下大家就会很快理解了。

当本地 DNS 服务器解析无法解析某一地址，比如 www.kkk.com，它会向上游 DNS 服务器发起查询从而获得一个参考回复，但反馈的结果可能是 ccc.com 域的记录，显然并不是想要的，此时如果启用此选项，DNS 服务器则不会缓存 ccc.com 域的相关记录，而只缓存 kkk.com 域的记录，这样做的目的是可以防止来自非法计算机冒充其他服务器发出的错误答复的干扰。在英文版系统中，这项设置被称为 Secure Cache Against Pollution，有些地方称为 DNS 中毒，说的就是这个意思。最明显的表现是明明输入的是 www.google.com，结果自动跳转到一个陌生甚至是 hacker 的钓鱼网站，或者是解析某个 URL 时，得到一个奇怪的 IP 地址，比如 12.34.56.78（此 IP 真实存在），类似这些情况，可能原因就是本机甚至本地的 DNS 中毒了。

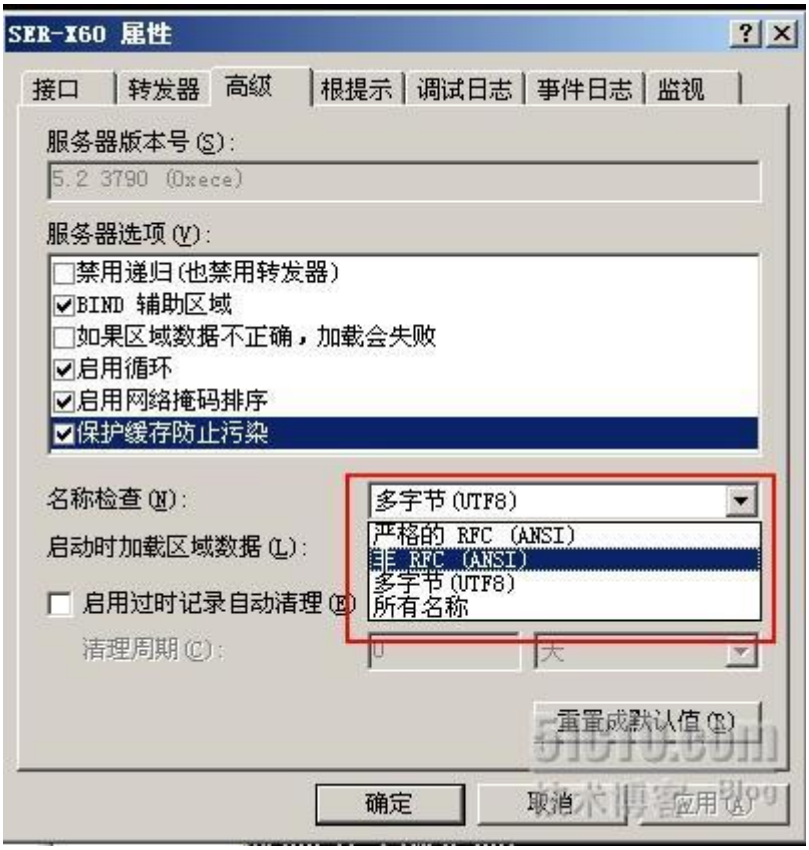
系统为了安全起见，这个设置也是默认被开启的。

第三部分：

这部分主要有 3 个选项，具体讲解如下：

【名称检查】

顾名思义，就是对提交的内容进行名称上的筛选，我们可以有 4 种选择，如下图：



默认是第三种，多字节（UTF8），那这几种有什么区别呢？如下图介绍：

方式	描述
严格的 RFC（ANSI）	按照 RFC 1123 中的定义进行名字检查，只能使用大小写字符（A～Z，a～z），数字（0～9）和连字符（-），DNS 域名的首字符可以是数字。
非 RFC（ANSI）	允许非标准的名字，并不遵照 RFC1123 规范，但是仍然使用 ANSI 字符。
多字节（UTF8） 默认方式	**允许使用 Unicode 字符作为 DNS 域名，从而允许使用非英文字符作为 DNS 域名，但是经过 UTF-8 编码后的域名长度不能超过 RFC 2181 中的定义长度（每节 DNS 域名不能超过 63 字节，完整 DNS 域名不能超过 255 字节）**
所有名称	允许所有的名字类型

我们平时看到的域名长度的一些规定，比如“每节 DNS 域名不能超过 63 字节，完整 DNS 域名不能超过 255 字节”，就是从这里来的。

这个设置我们基本不去理会，了解就好。

【启动时加载区域数据】

这里有三个启动选项，如下图：

默认情况下 DNS 启动时是从 AD 和注册表中加载启动文件，对于非与环境，自然只能从服务器的注册表中加载了启动数据了。“从文件”这项是指 DNS 会类似 BIND 一样依靠 named.boot 这样的文件进行启动的设置，此设置保持默认即可。

【启动过时记录自动清理】

如果想定期清理 DNS 的一些记录，如无效缓存或过时的记录等就可以通过设置清理周期来完成。

此选项卡右下角的【重置成默认值】是可以讲设置还原到最初状态，当设置出错或打算清除以往设置时，点一下该按钮即可恢复如初。

以上是 DNS 服务器属性的【高级】选项卡的全部内容，东西很多，知识点很丰富，希望可以对你有所帮助。

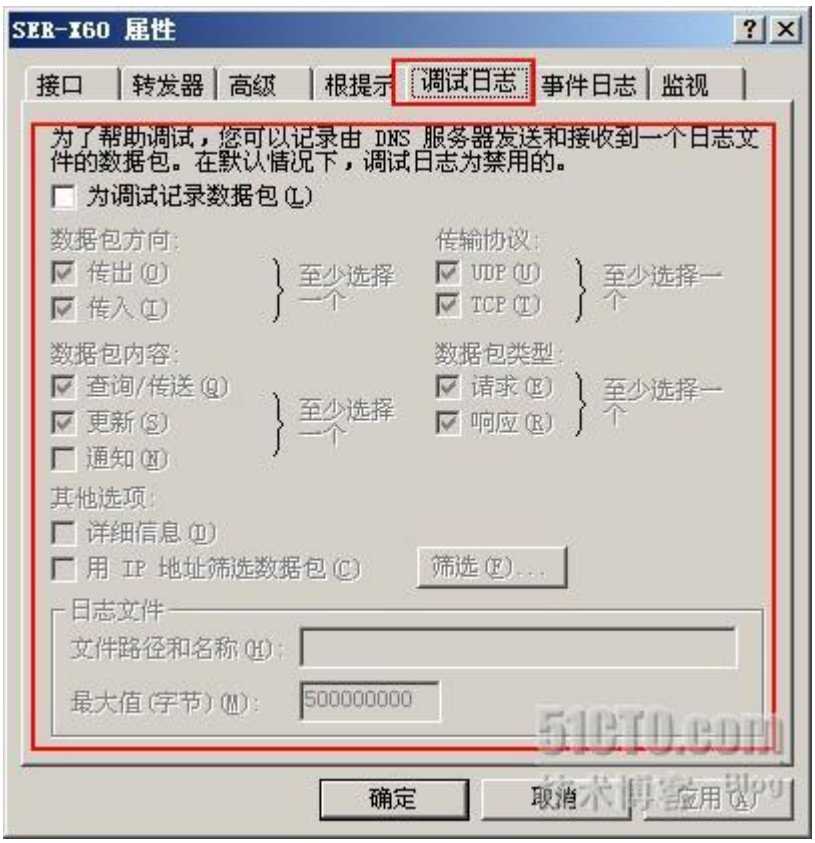
剩下的几个选项卡会在下面的章节做介绍，敬请期待！

谢谢！

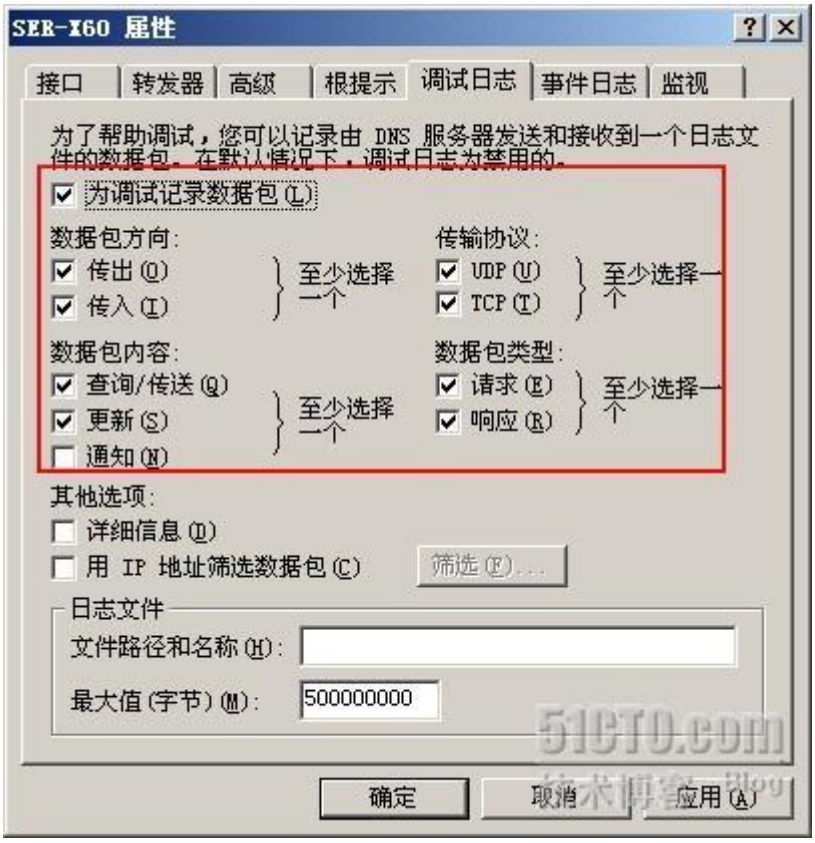
一起学 DNS 系列（十二）DNS 服务器属性之调试和监视选项

接着第九节的部分，一起来探讨 DNS 服务器的剩下几个选项，内容比较少。

如下图：

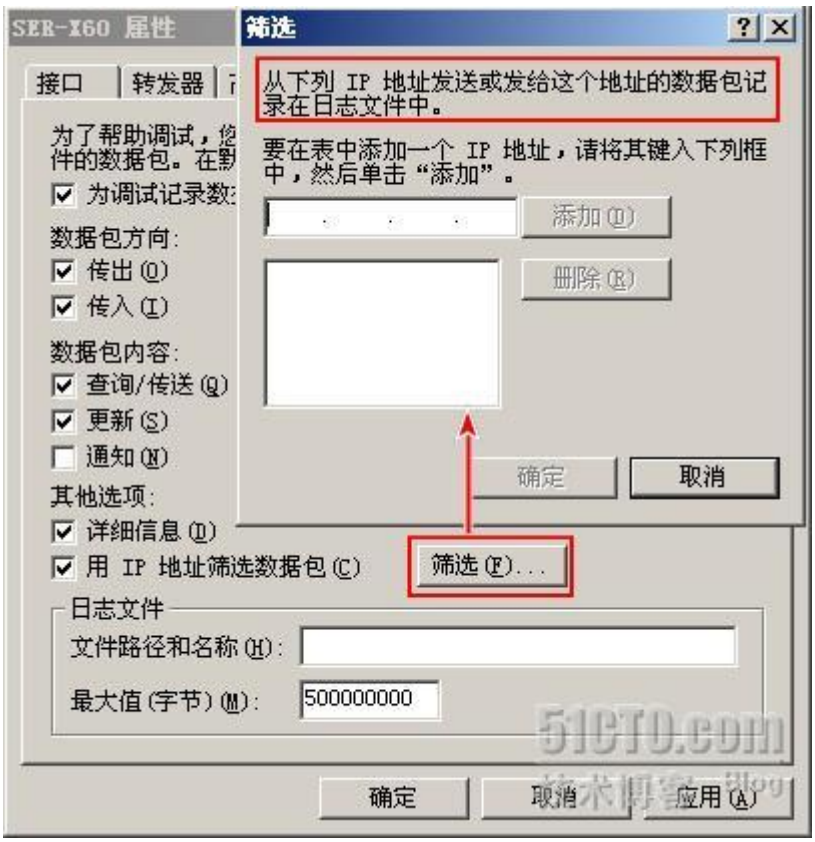


这里，我们可以对 DNS 数据进行有选择的记录，通常这些都是用来辅助解决问题的调试日志，由于当 DNS 查询和请求连接很多时，开启调试功能会对服务器性能造成一定影响，所以默认是被禁用的。我们也可以手工开启。如下图：



调试日志分为 4 大类，每一类必须要选择一项，具体分类就不多说了。在【其他选项】中，我们可以通过筛选功能监视某一个或一些的 DNS 数据包的情况。

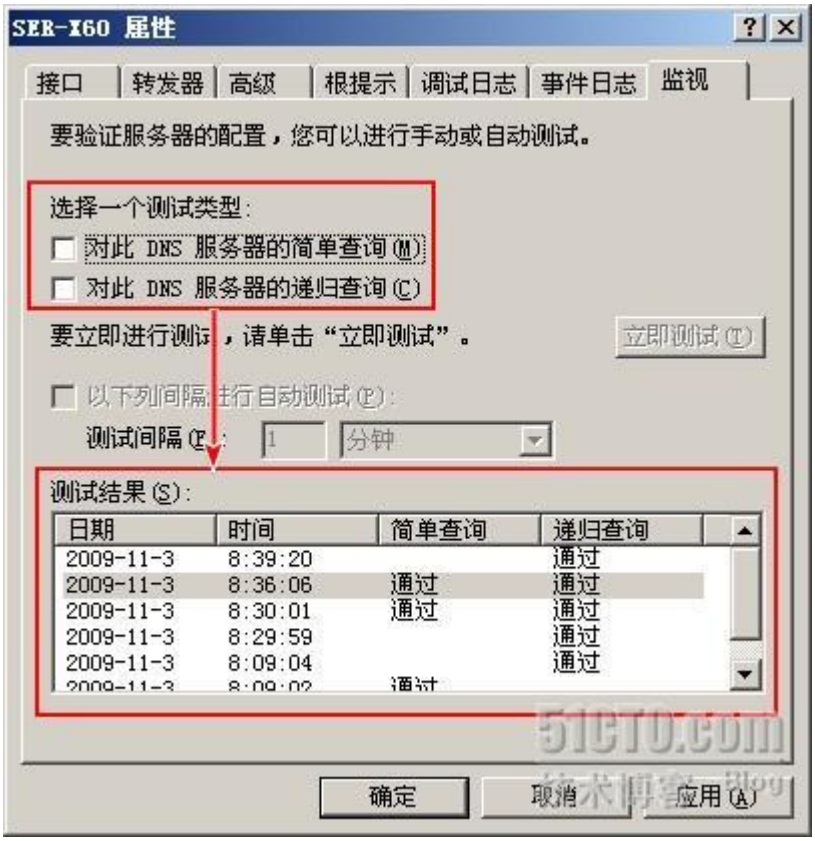
如下图：



【日志文件】里，默认情况下如果不输入日志保存的文件名称。系统会自动保存在 C:\WINDOWS\system32\dns，这个路径下，文件名为 dns.log，这里也可以自定义一个名称，同样会保存在 dns 目录下；我们也可以键入绝对路径，比如 D:\dnslog\dnslog.txt，类似这样的也可以，**如此同时原目录下的 dns.log 文件将会自动清除。请大家留意此处。**

【事件日志】没有什么好说的，通常我们可以通过事件查看器工具同时浏览到 DNS 的事件记录。

【监视】选项卡，其实是对 DNS 服务器性能的一个测试和监控。通过这个选项卡，我们可以完成对此 DNS 服务器的简单查询和递归查询的测试。如下图：



简单查询属于 DNS 内部查询，不会涉及到往网，所以通常这个测试很容易通过。但 DNS 服务器的递归查询需要测试到根提示服务器的连接性，如下图：

DNS服务器IP

针对几个根服务

如果查询不通过，那么就需要检查一下根提示服务器列表是否有问题，再就是查看 DNS 的网络连接是否正常。

有关 DNS 服务器属性的部分就讨论到这里，本节内容很少，也很好理解。

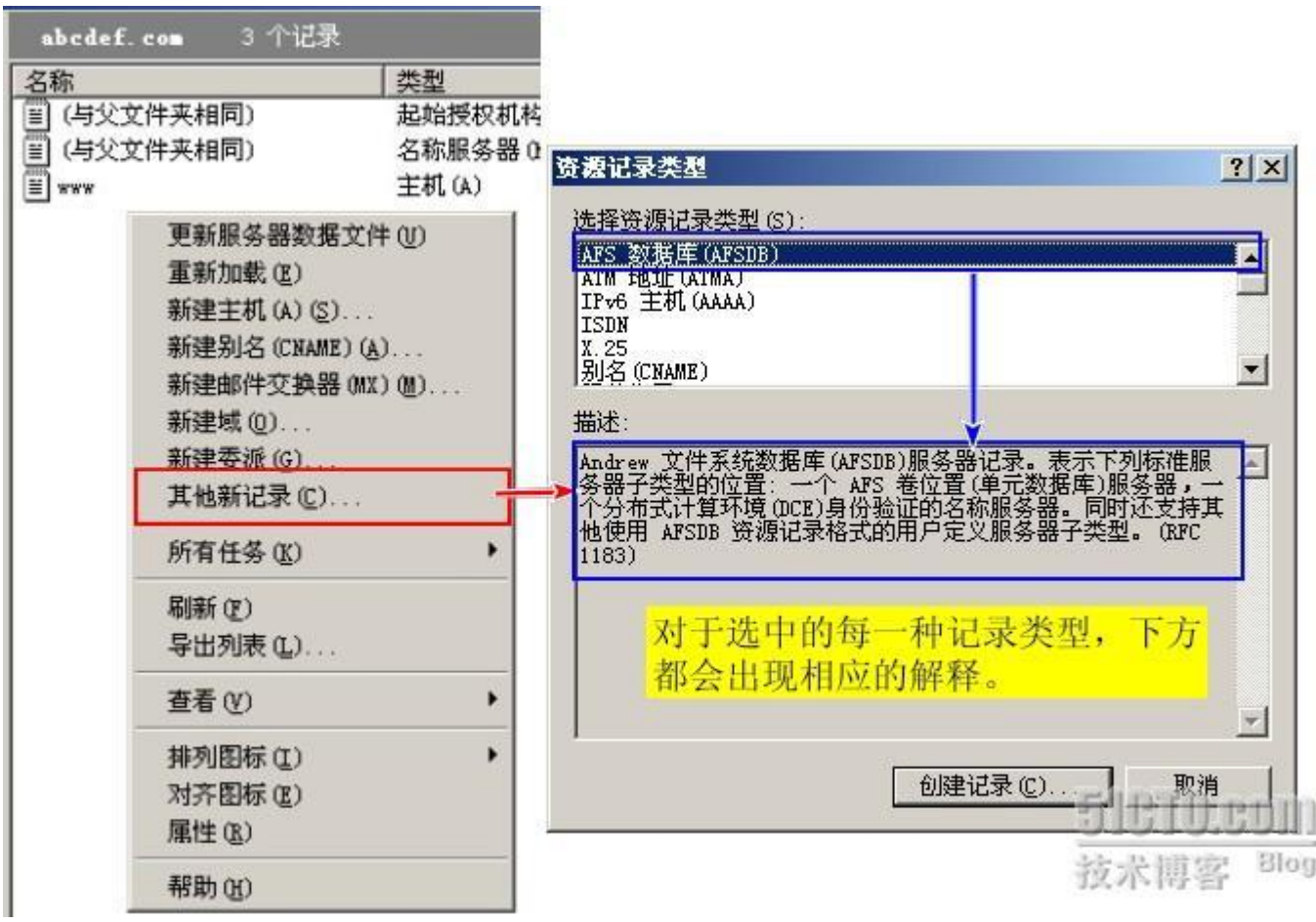
谢谢大家！

一起学 DNS 系列（十三）图文详说 A、CNAME、MX 和 NS 记录

前面用了 12 个小节对 DNS 的基础、以及 Windows 的 DNS 系统作了较详细的描述，下面的几节主要是说一些有关 DNS 应用方面的内容。

DNS 记录

如果把 DNS 的体系结构比喻成一棵倒挂的大树，那么毫无疑问，每一条记录就是组成这棵大树必不可少的枝叶了。所谓的 DNS 记录，其实就是具有特殊功能的一个个数据条目。在 windows 的 DNS 中，这些条目一旦被创建后，就可以实现各式的功能，比如创建一条 A 记录，就可以为客户端提供某个域名到 IP 的正向解析功能等。当然，DNS 记录分为很多种，各有各的用途。进入 DNS 管理系统，我们可以在这里看到。如下图：

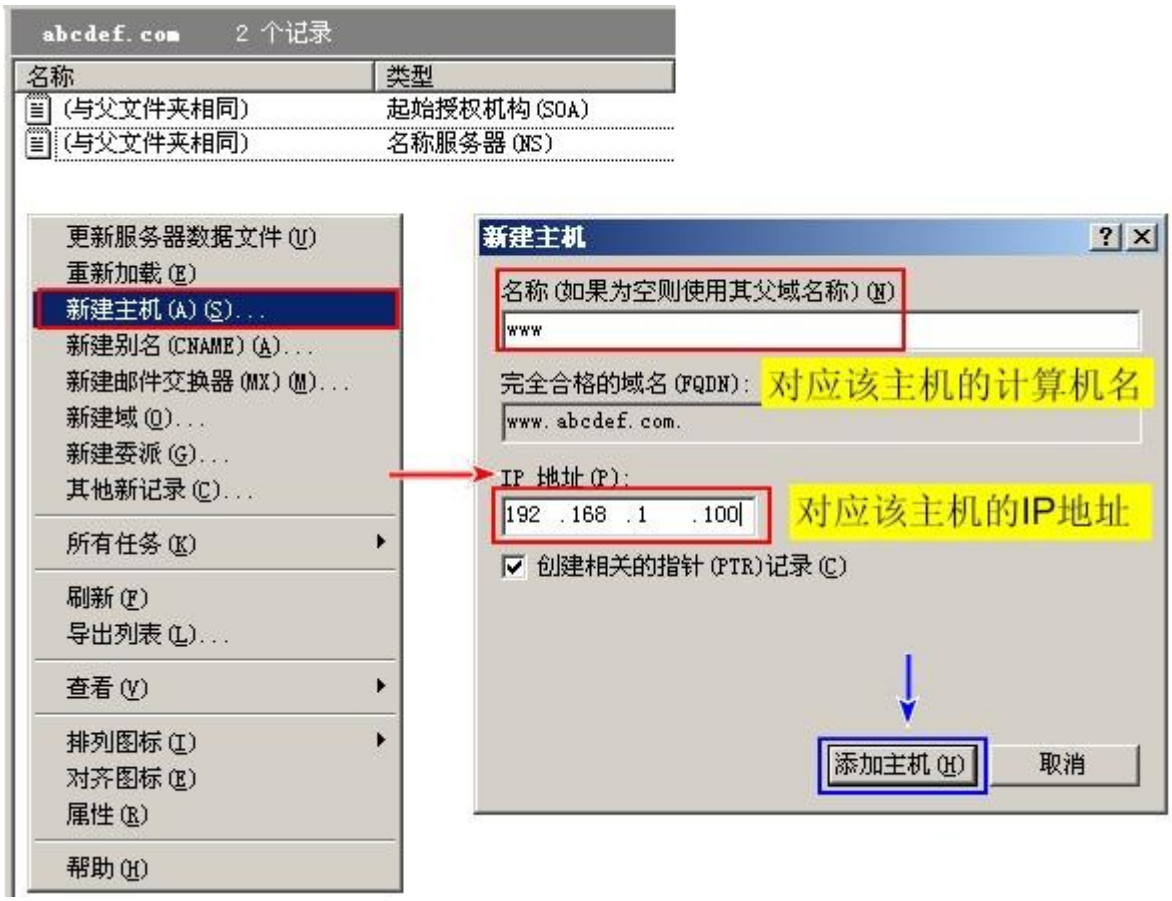


尽管这里列出了几十种 DNS 记录，这里只说说我们接触较多几类记录。

A 记录

当我们想获取一个域名对应的 IP 地址，或通过域名方式访问某一网页或程序，此时就需要在这个域名和所属的 IP 地址间创建一个映射关系。这个关系就是利用在 DNS 中为此名称创建的 A 记录。而这个名称可以理解成是某台主机的计算机名如 www，它的 IP 是 192.168.1.100，同时，在这台主机安装 IIS 并创建一个测试页面。当 DNS 服务器上存在一个 abcdef.com 的区域，同时，将 www 这台主机的主 DNS 后缀设为 abcdef.com，现在，我们想在局域网内实现

通过 www.abcdef.com 就可以访问那个测试页面，那么就需要在 DNS 上做一个 A 记录，目的是把 www.abcdef.com 和 192.168.1.100 对应起来，如下图：



按照上述步骤创建完成后，如下图：

在列表中会出现一条 A 记录，同时这个条记录的对应的 FQDN 就是 www.abcdef.com，我们来验证一下创建的结果，如下图：

因为试验环境里没有 192.168.1.100 这台计算机，所以 ping 无法通过，但如果真实存在的话，那就不会有什么问题，同时访问测试页面也会很顺利（此处不再演示）。我们可以为一个域名添加多个 IP，同一 IP 也可以对应多个主机名。这样做的目的是可以实现简单的冗余访问。

以上是本地 DNS 的 A 记录操作方法，如果你有一个付费域名，想让用户通过它来访问某个网站，那么就需要在该域名的控制台上添加 DNS 记录，这里以易名中国为例，进入域名管理界面，如下图：

上图有一个域名 zenter.cn，点击右侧的[管理]，就可以对这个域名做进一步的操作。

如下图：

上图是这个域名的概览页面，点击下方的[解析管理]就可以添加或删除 DNS 记录了。

如下图：

按照上述文字说明，主机名为 www，IP 地址为 192.168.1.100 即可。这样就创建了一条 A 记录，当访问 www.zenter.cn 时，DNS 服务器会自动解析到 IP 为 192.168.1.100 的主机。总的来说，A 记录即 address 记录，目的是标识出一条特定的域名到 IP 地址的记录。

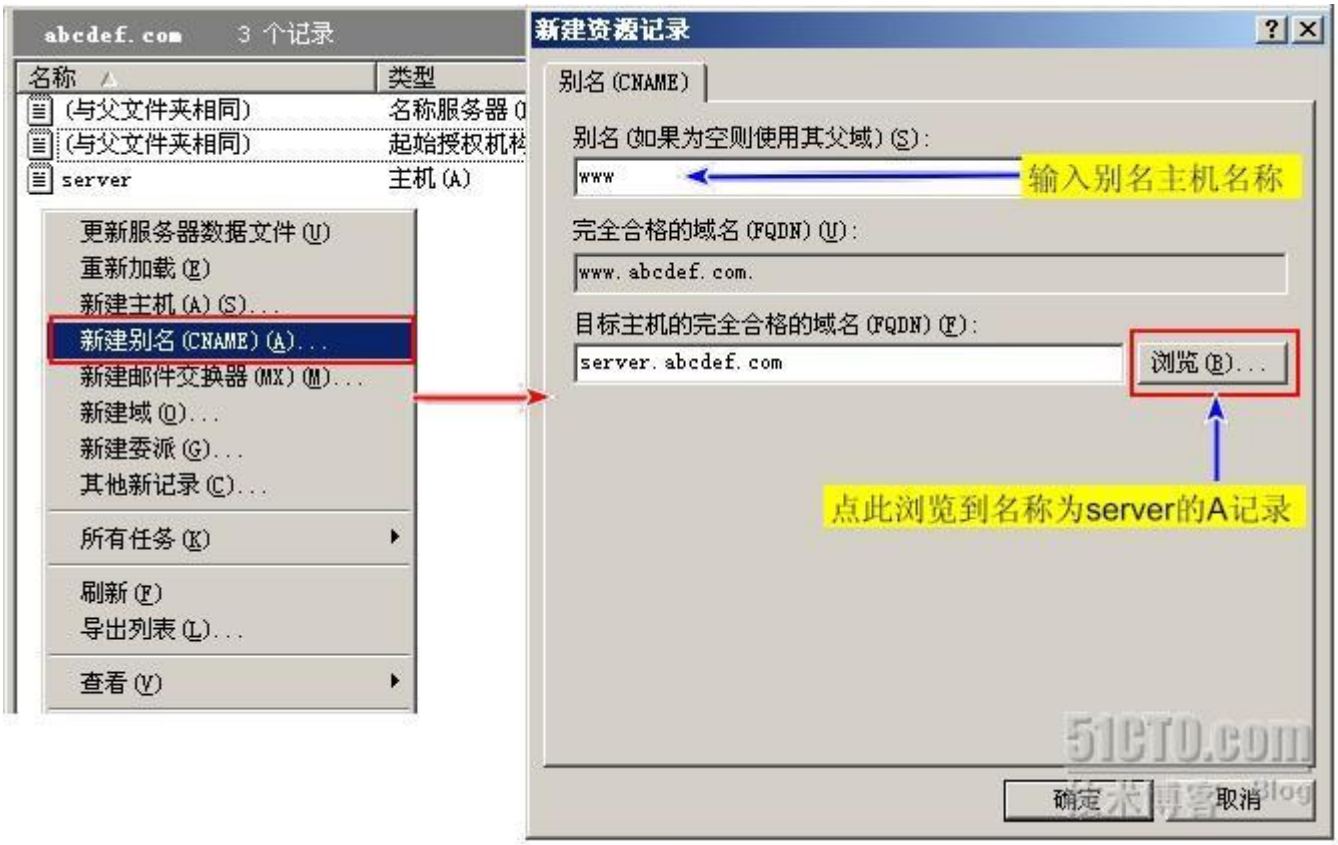
CNAME 记录

CNAME 记录，即别名记录。我们通过设置别名记录，可以将多个名称指向同一台服务器。比如有台名为 server 的主机上提供邮件和网页服务，我们可以设置 www 和 mail 这 2 个名称的别名记录指向这台服务器，用户可以通过 www.zenter.cn 和 mail.zenter.cn 来访问各自需要的服务，但实际上目标都是同一台服务器。

这里做个演示，如下图：

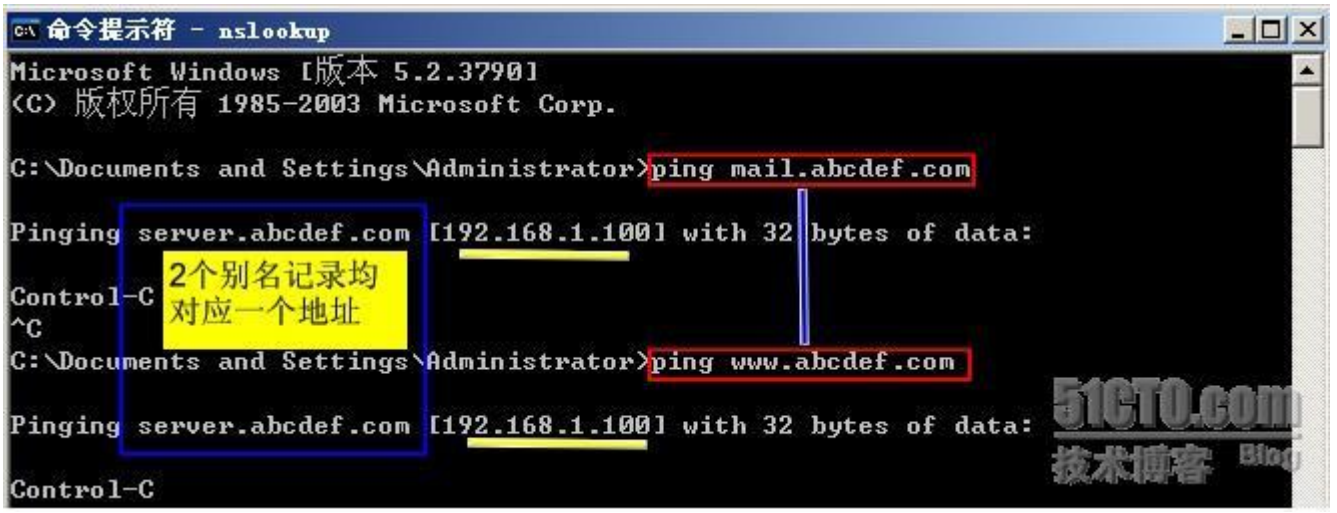
首先建立一个 A 记录，这个是创建 CNAME 记录的基础。

A 记录创建完成后，继续下一步，如下图：



我们选择【新建别名(CNAME)】，创建方法如上。创建完成后，如下图：

请大家留意上图中的 DNS 记录的类型。创建完成后通过命令来验证一下，如下图：

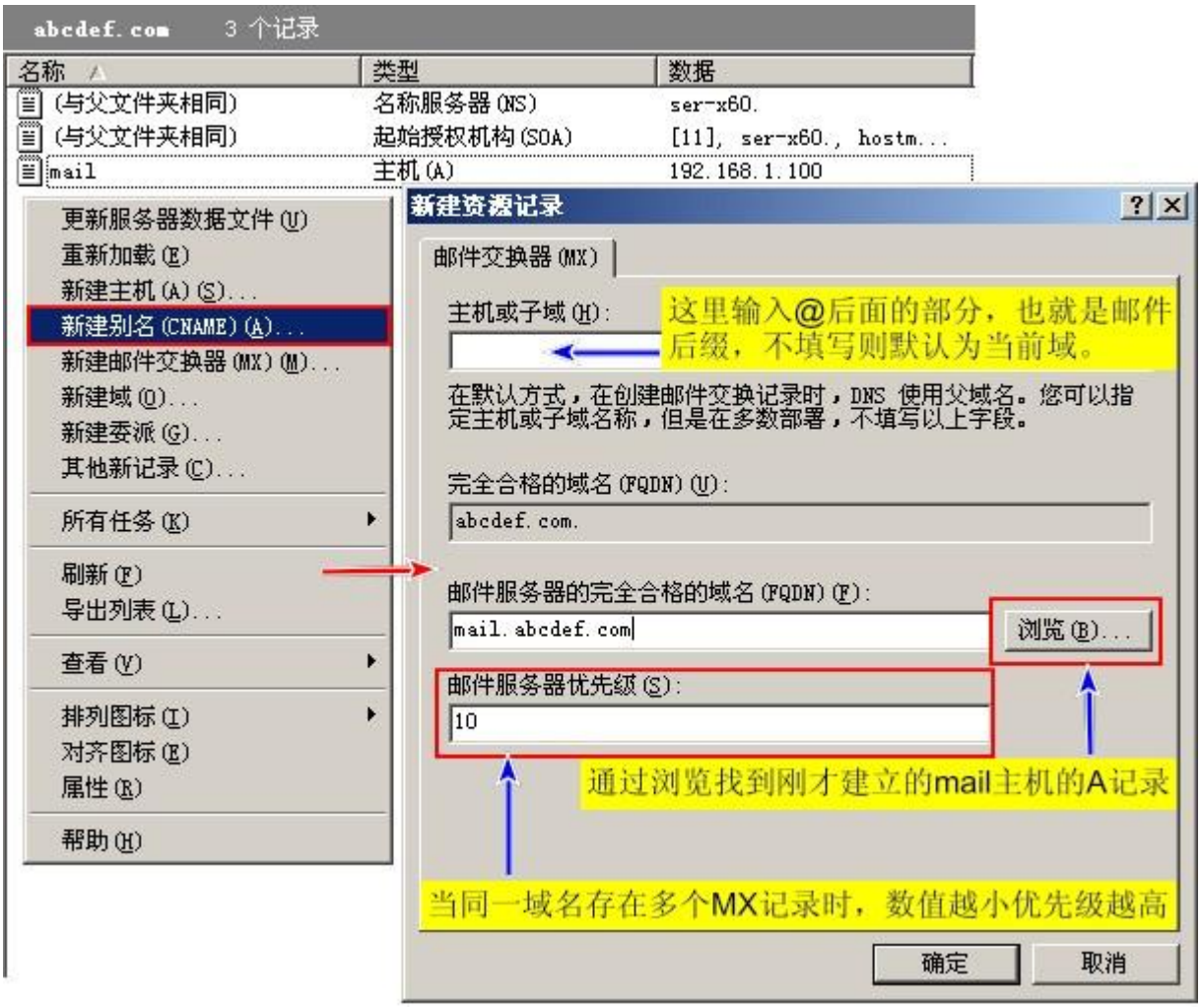


对于 CNAME 记录如何转换到 A 记录上这个问题，我们不用深究，这个过程是在 DNS 内部实现的。

MX 记录

MX 记录即 Mail Exchanger，主要用于邮件服务器，作用是用于定位邮件服务器的地址。如一个用户给 user@abcdef.com 的用户要发封邮件，此时该用户的所属的邮件系统会通过 DNS 服务器来查找 abcdef.com 这个域名的 MX 记录，如果存在，就会根据这个 MX 记录来查找对应的 A 记录，从而得到邮件服务器的 IP 地址，并将这封邮件发送到这台服务器上。可见，MX 记录和 A 记录是分不开的。总的来说，MX 记录是为了让对方找到你的邮件服务器，所以，如果想顺利收信，就必须为你的邮件服务器创建合法有效的 MX 记录。

我们现在给 mail 这个主机创建一个 MX 记录（A 记录创建过程省略）。如下图：

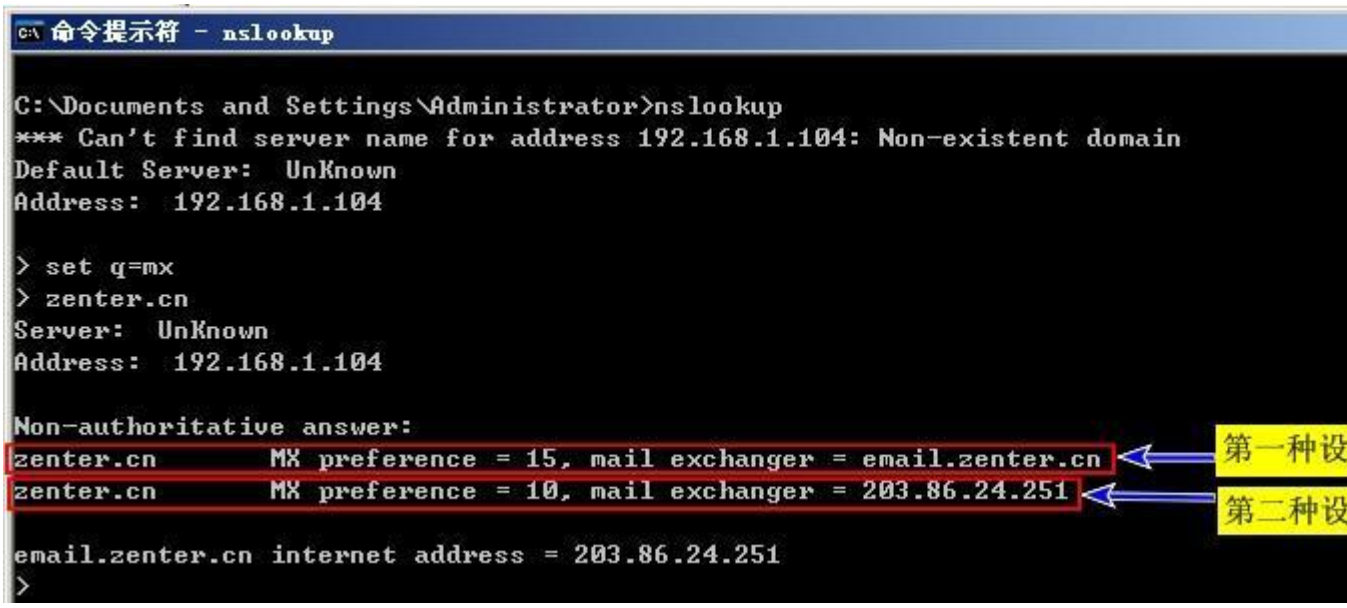


其实，如果新建一条主机名为 email 的 A 记录，只要和主机 mail 指向的 IP 一样，再在这个基础上做 MX 记录，效果是一样的。也就是说，MX 记录所对应的 A 记录的 IP 一定要是你的邮件服务器的 IP，这样才可以被外部邮件系统正确识别。如果有多台邮件服务器，并已组成集群，然后为每一个服务器都创建一个 A 记录和对应的 MX 记录，此时每个 MX 记录就可以使用不同的优先级了。

依旧以易名网上的 zenter.cn 域名为例，来看一下在域名控制台上如何做 MX 记录。

如下图：

那这两种设置方式不同会有什么异同呢？我们用 nslookup 命令检查一下，如下图：



其实这个差别是很小的，在查询的时候就可以看到了。通常情况下建议以主机名为主，因为并不是每个域名商都允许这样设置，比如万网，在做 MX 记录时，只允许输入域名，而不能是其他值。如下图：

所以建议按照规范来创建 MX 记录。

NS 记录

之所以最后讲 NS 记录，是因为平时我们几乎不用去创建 NS 服务器，因为大多数域名商默认用自己的 NS 服务器来解析用户的 DNS 记录，当然，如果你可以自建 NS 服务器。不过前提是，你需要在本地 DNS 服务器上创建好 NS 记录，并将此 DNS 服务器 IP 告之对应域名商，只有他们将此 IP 登记到互联网上后，本地的 NS 服务器才可以正常解析 DNS 请求。

但无论怎么样，首先必须要有一个合法的域名，这一步是不可或缺的。以 zenter.cn 为例，然后搭建一个 DNS 服务器，可能用 windows 的 DNS 或 Linux 下的 BIND。然后创建了 2 条 NS 记录，ns1.zenter.cn 和 ns2.zenter.cn，它们对应的 IP 都是 1.1.1.1。然后我将此 IP 地址告之**易名中国**，他们会将这个 IP 在互联网中心注册，大约 48 到 72 小时后就可全球生效，这样我这台 DNS 就可以创建 A 记录、MX 记录等了。也就等同于，这台 DNS 服务器是面向公网服务的。

那么，我可以用这个 NS 服务器用来解析其他的域名，要做的只是将域名商默认的 NS 服务器替换成 ns1.zenter.cn 和 ns2.zenter.cn。如下图：

本节较详细的描述了有关 A 记录、CNAME 记录、MX 以及 NS 记录的内容，希望对大家能有所帮助。

一起学 DNS 系列（十四）DNS 查询工具之 DIG 的使用（1）

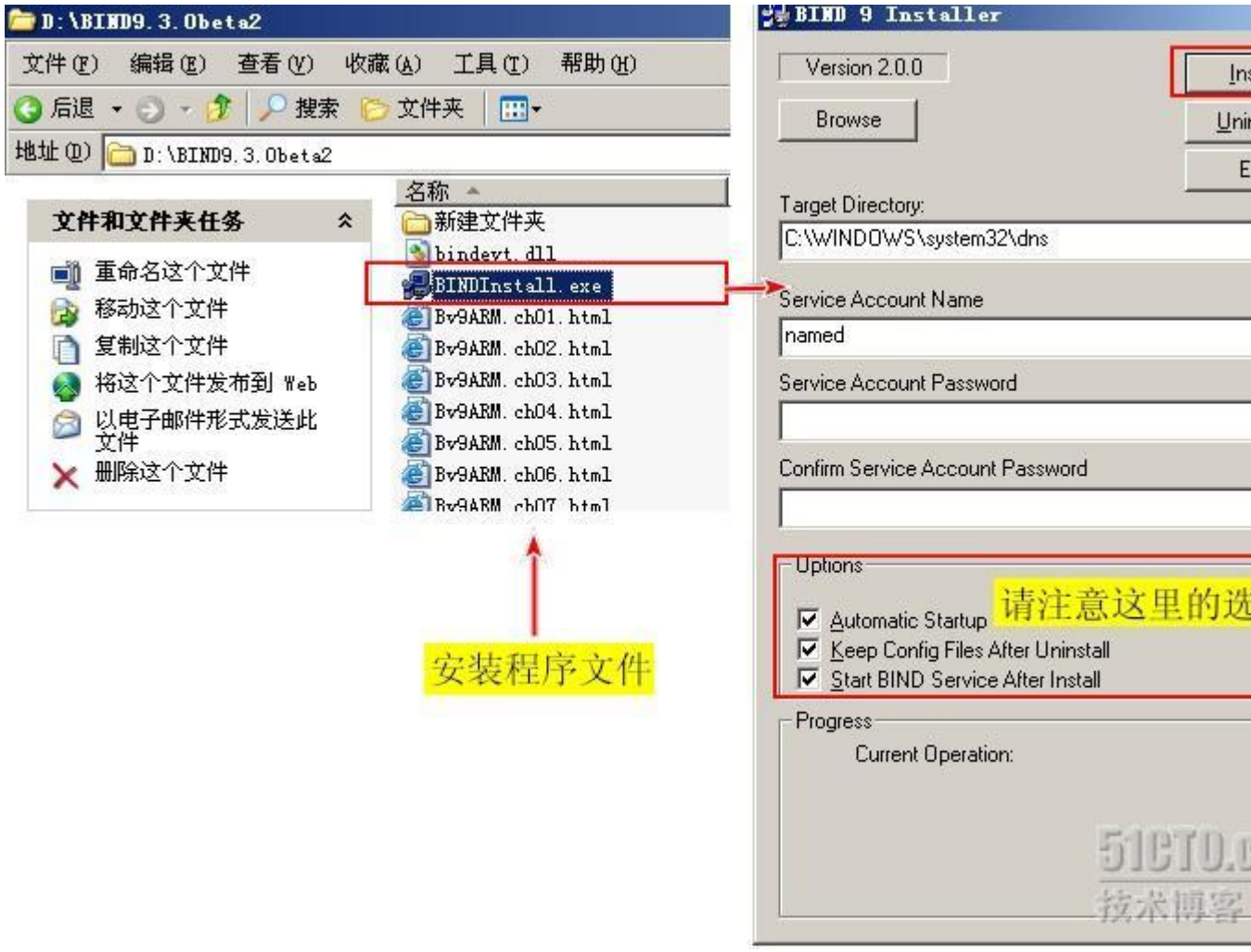
这几节主要讲 2 个常用的 DNS 的命令行诊断工具，分别是 linux 下的 DIG 和 windows 下的 NSLOOKUP，我们首先来说 DIG 工具。

DIG，全称 Domain Information Groper。原本是 Linux 平台上 BIND 服务器诊断的工具（已上传至附件），现在已经有了 windows 的版本，这节就是利用此工具在 windows 环境下来做一些 DNS 的诊断测试。

首先，需要将 DIG 安装到系统中，大致分为以下几步：

1、安装 DIG 程序

运行安装程序中的 BINDInstall.exe，如下图：



运行完成后，会弹出一个对话框，告诉我们安装已完成。如下图：

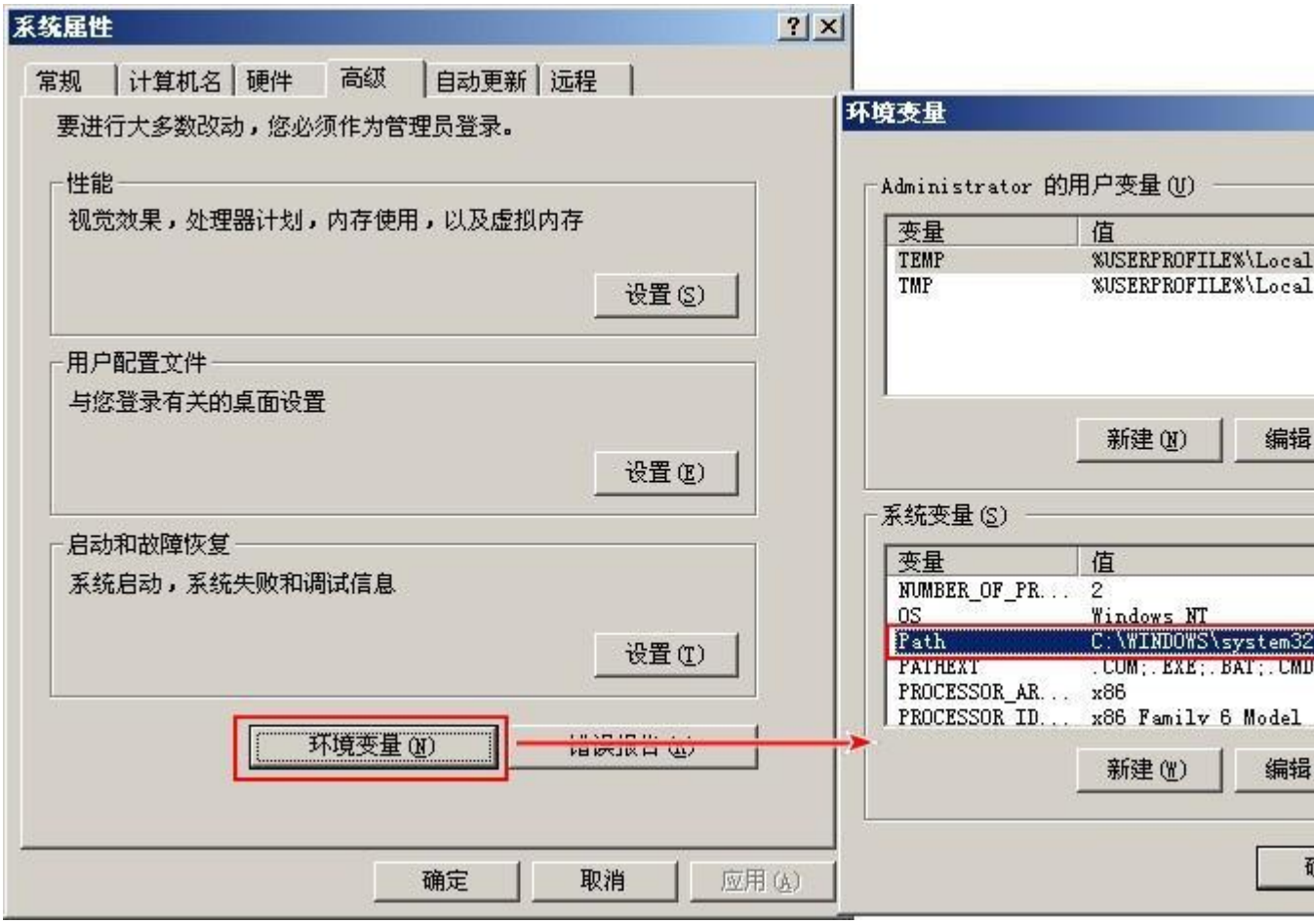
程序默认将关键文件安装到 C:\WINDOWS\system32\dns 下的 BIN 文件夹下，如下图：

红框的文件就是 DIG 程序。至此，程序安装完成。

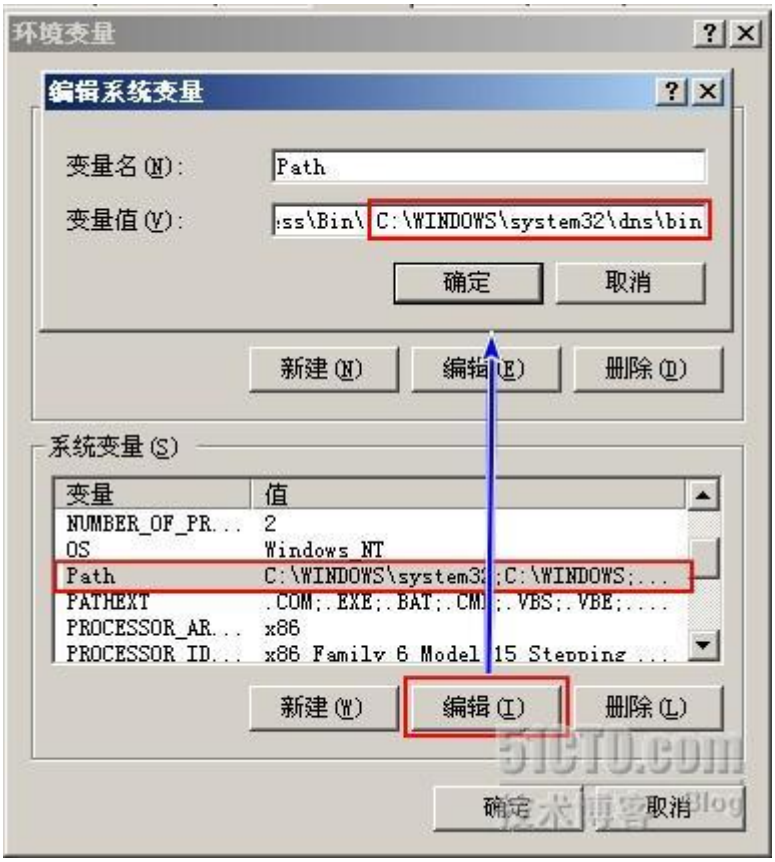
2、设置 DIG 运行环境

待安装完成后，我们需要将上述安装目录添加到环境变量中，以方便在命令行中直接运行 dig 命令，而无需使用绝对路径。具体步骤如下：

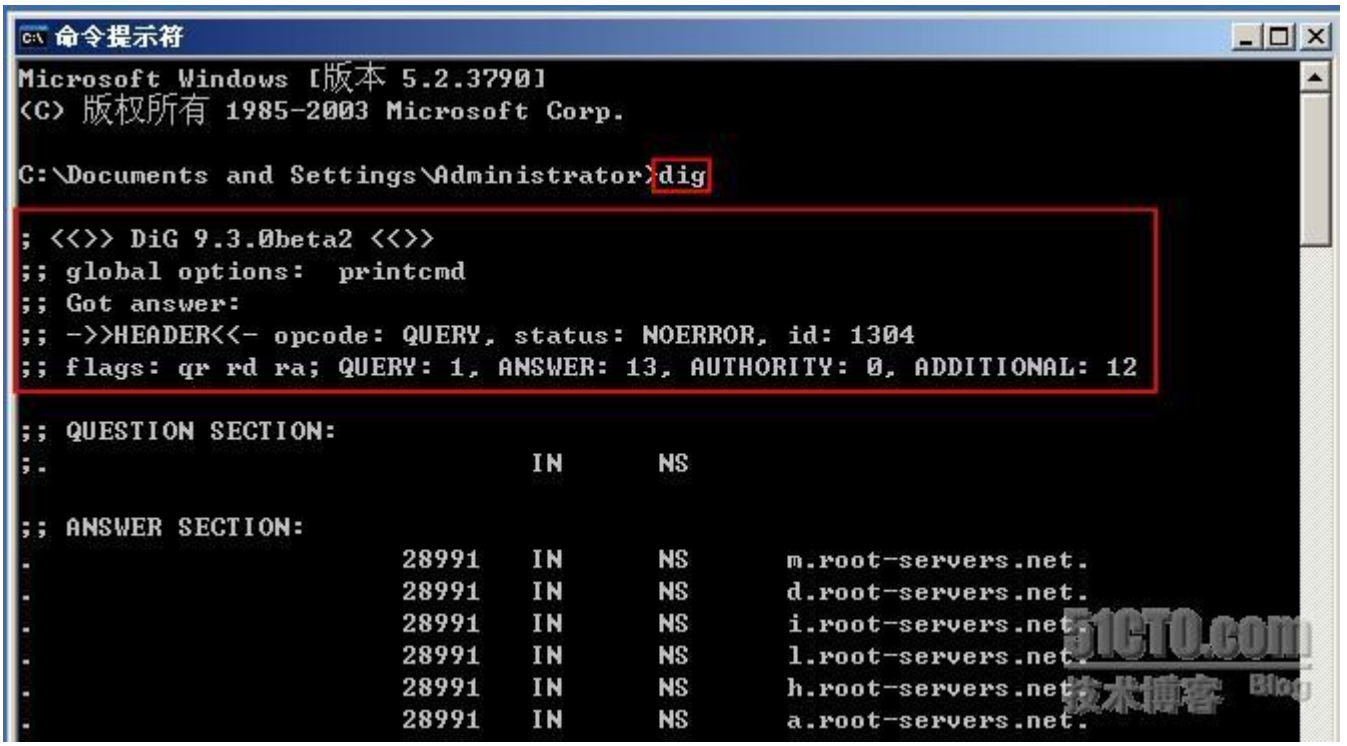
打开系统属性，并找到高级选项卡。如下图：



我们需要把 DIG 的路径添加到系统变量中，如下图：



添加完成后，在命令行里测试一下。如下图：

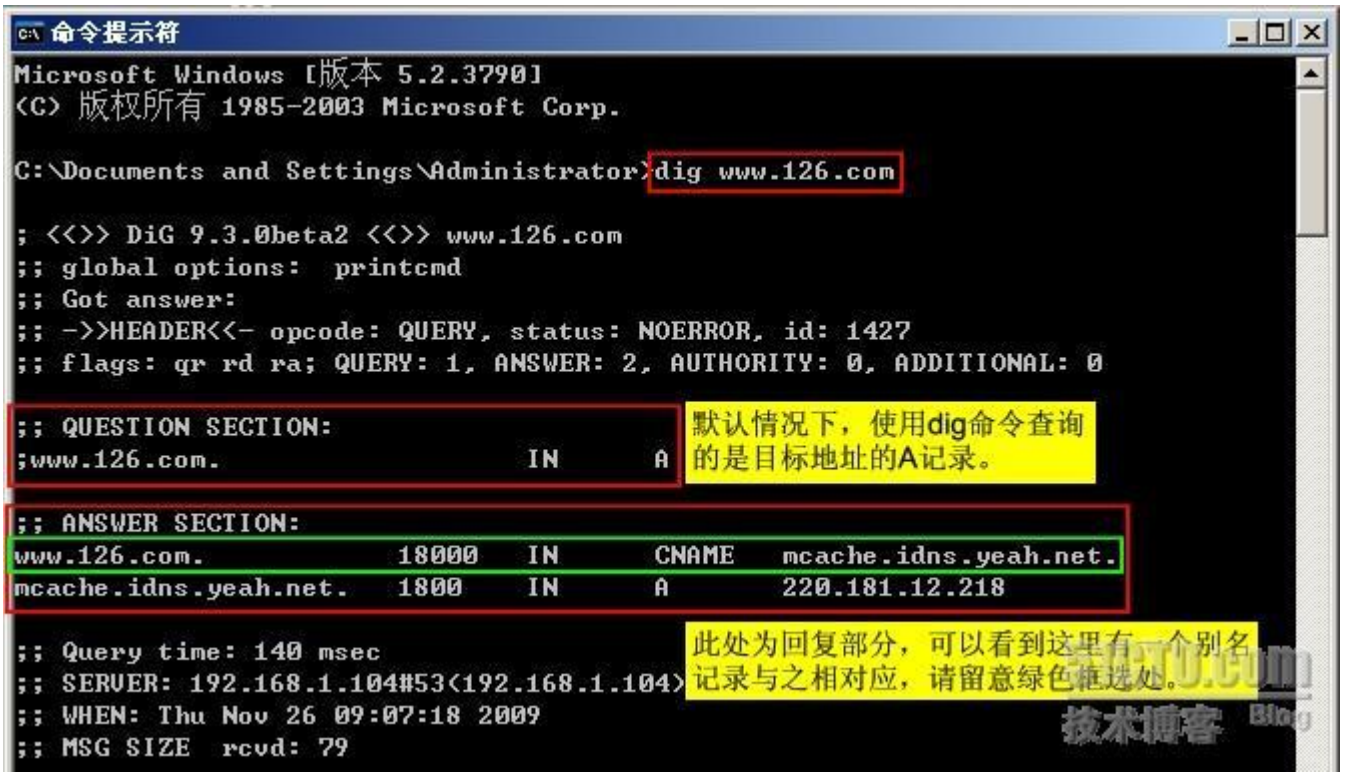


出现上图提示就表明 DIG 程序安装成功，下面对输出内容做简要介绍，如下图：

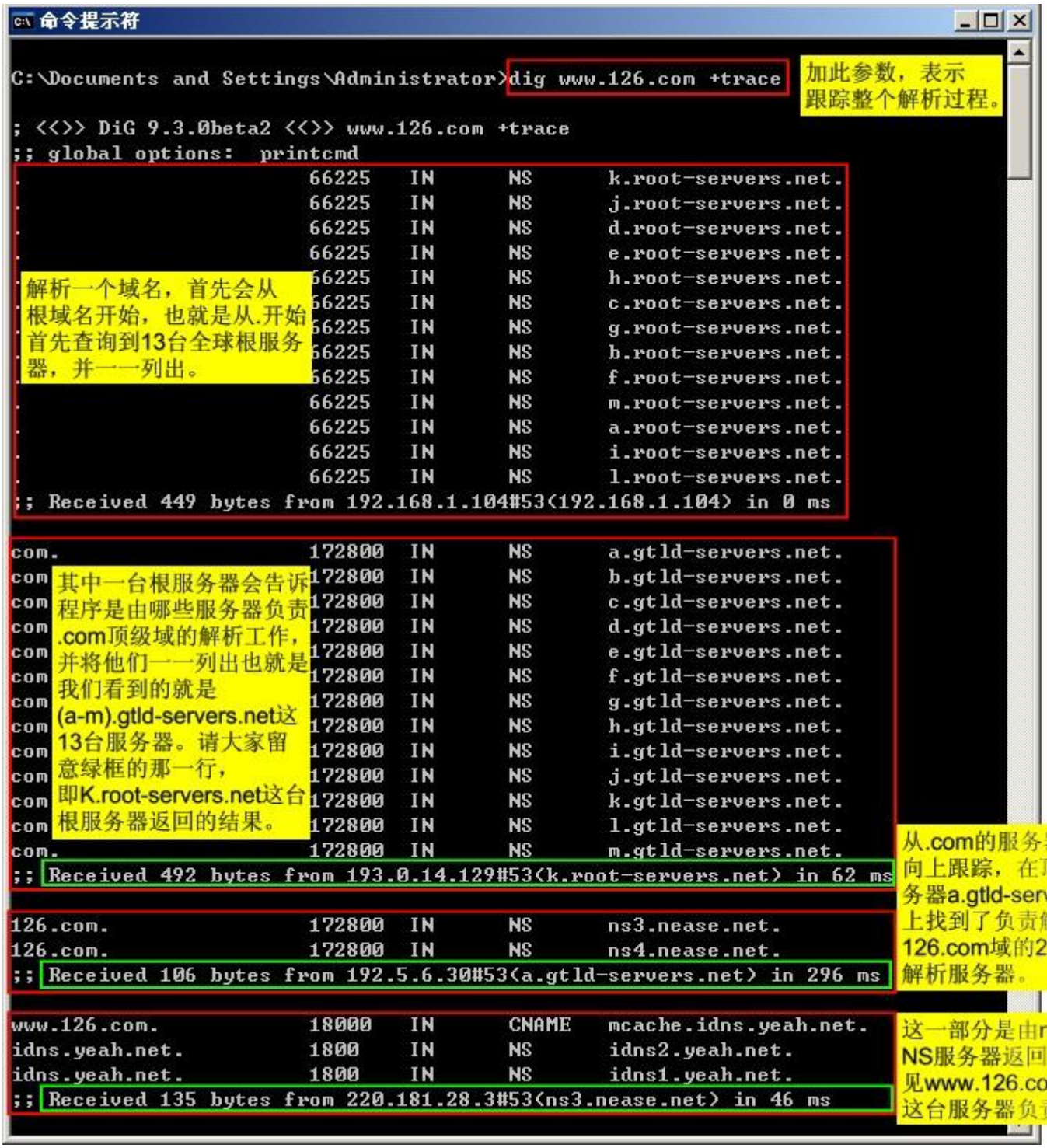
当不添加任何参数，直接使用 dig 命令时，默认会解析根服务器，并附加解析对应的 A 记录。

如果想解析一个域名，则之需要在 dig 后面添加域名地址即可。

以 www.126.com 为例，如下图：

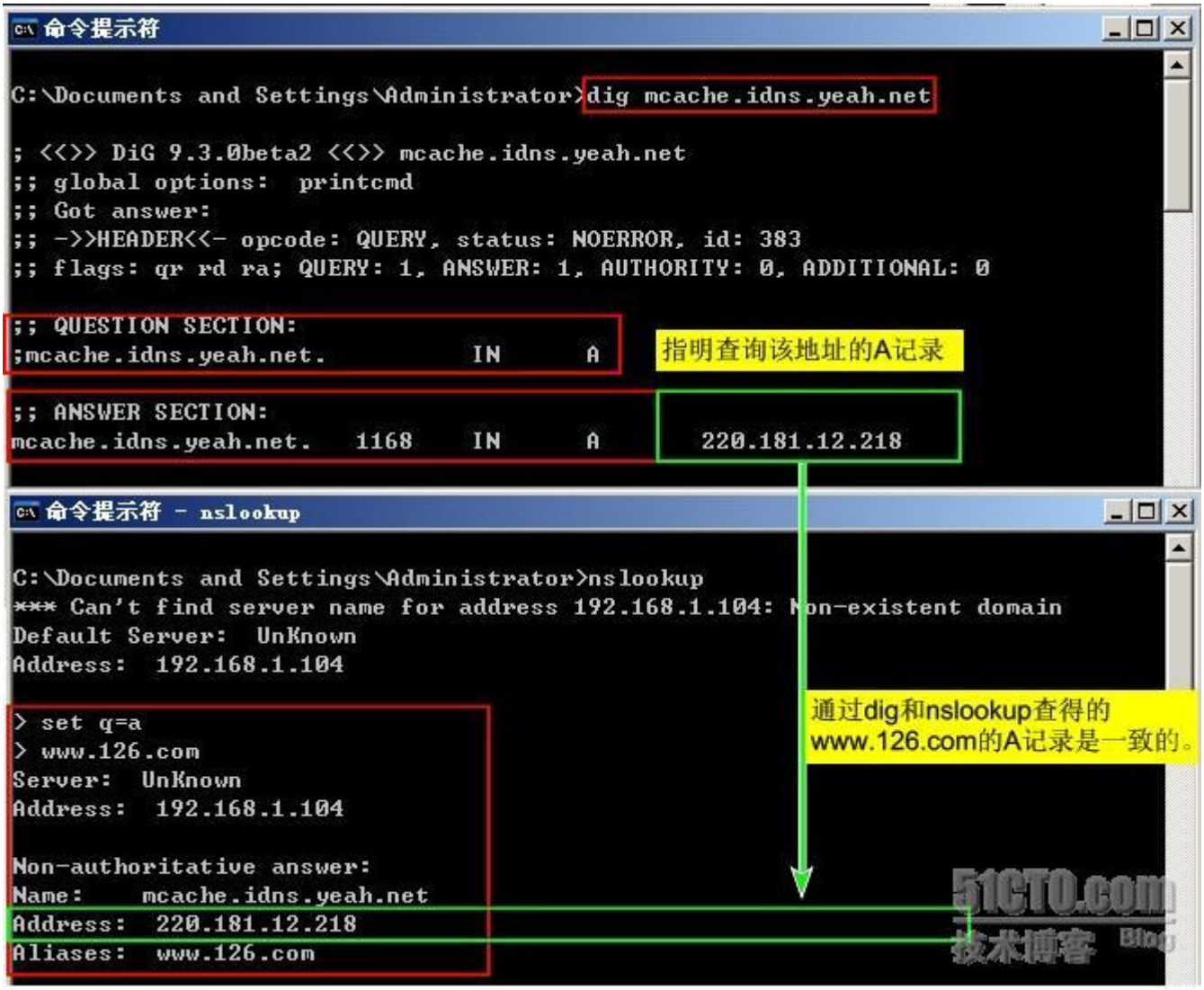


上图是利用 dig 命令查询 www.126.com 的 A 记录，但这样的结果过于简单，并没有体现整个解析过程，此时我们可以借助+trace 这个参数，来获得过多的信息。如下图：

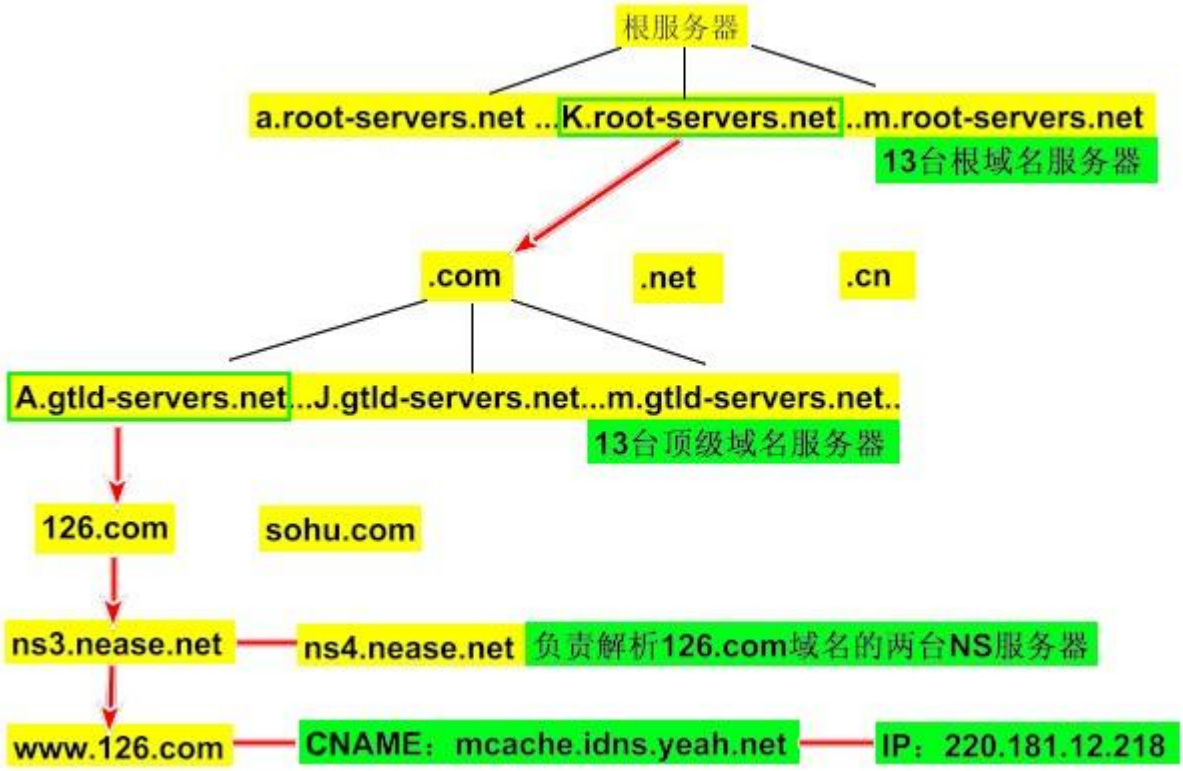


上图是利用 dig 命令解析 www.126.com 的 A 记录的整个过程，我们可以清晰的看到首先是查找根 DNS 服务器，然后其中的 K.root-servers.net 根服务器告诉 DNS 负责.com 域名的 13 台顶级域名服务器，接着由其中的 a.gtld-servers.net 顶级域名服务器告诉 DNS 负责 126.com 的 2 台 NS 服务器分别是 ns3.nease.net 和 ns4.nease.net，最后 DNS 继续查询负责解析 www 这台主机的 NS 服务器，结尾一行表明是由 NS3 这台 NS 服务器负责解析 www.126.com 这个地址的 A 记录，只是这个 A 记录做了别名记录，并未直接显示出来而已。

但我们可以继续利用 dig 命令查这个 CNAME 记录所对应的 A 记录。如下图：



上两幅图掩饰了 www.126.com 地址完整的解析过程。我们可以把这个过程以图表的形式体现出来，也就是整个解析路径，如下图：



从这张图大家应该大致了解整个解析的过程和步骤。

这 2 个命令仅仅是 DIG 工具极小的一部分，下节会继续介绍相关内容，敬请期待。

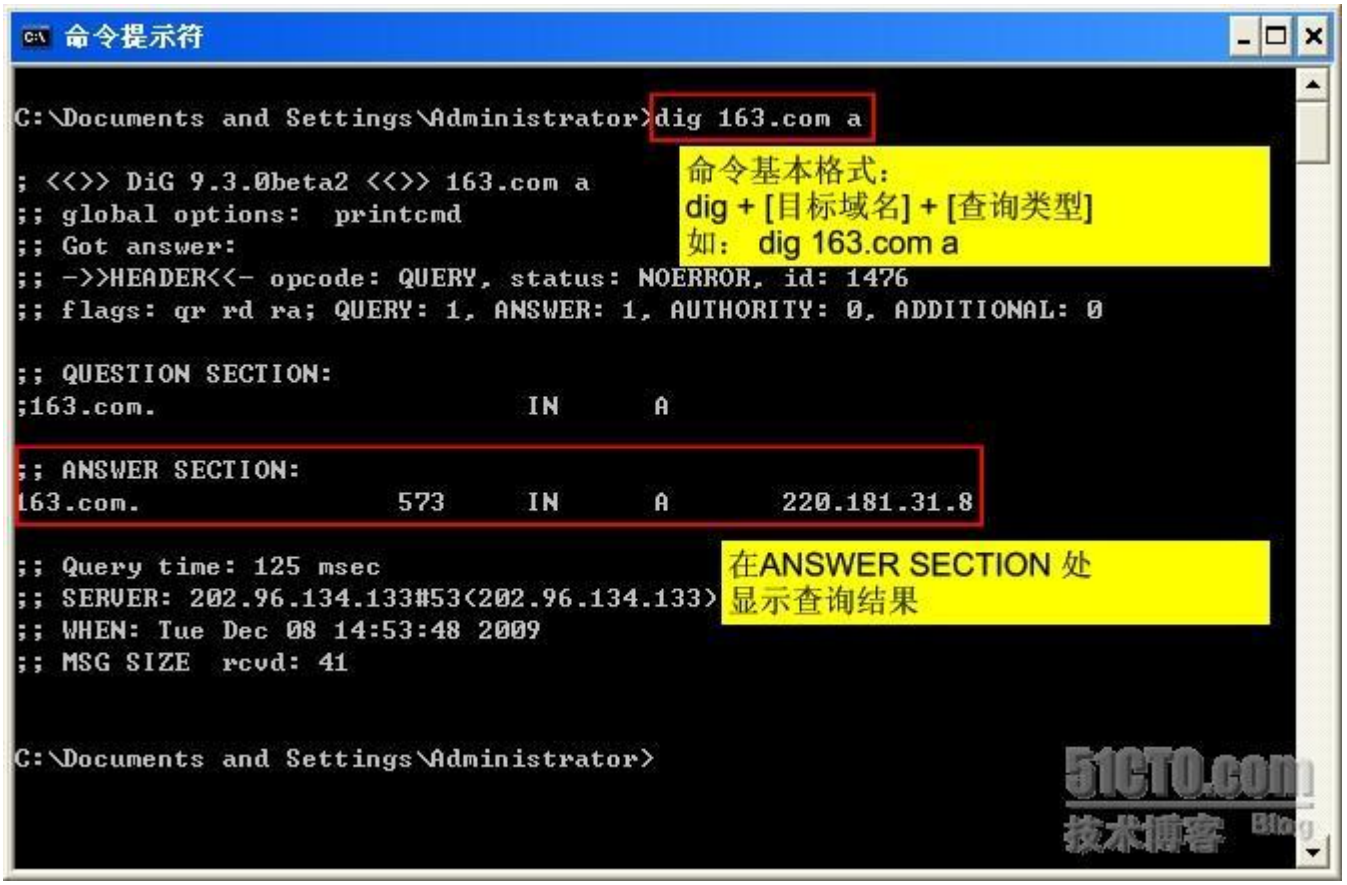
谢谢！

一起学 DNS 系列（十四）DNS 查询工具之 DIG 的使用（2）

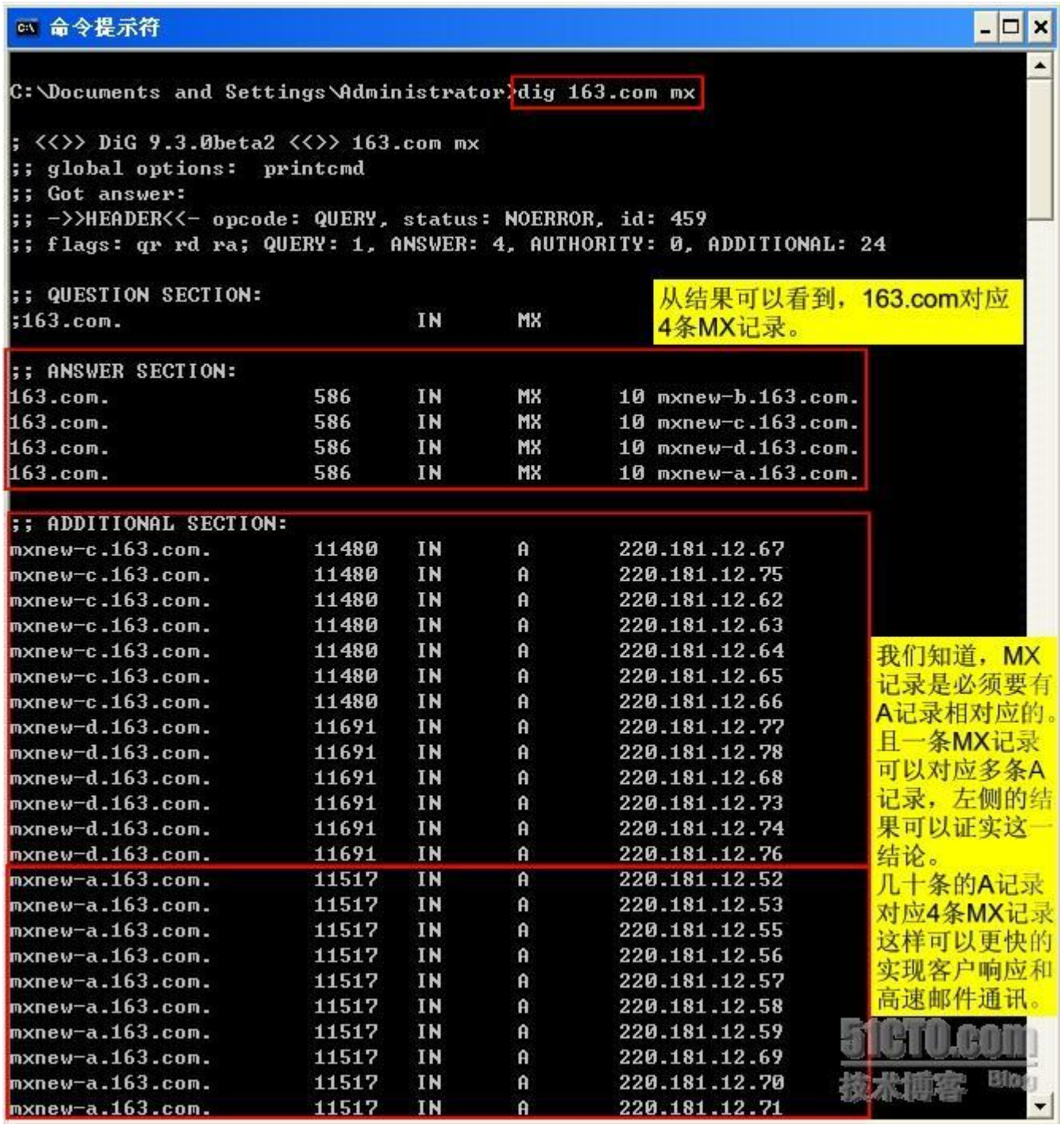
上节谈到了利用 DIG 的+trace 参数追踪域名解析的过程，其实除此之外，DIG 还有一些其他的参数，这节会涉及到这部分内容。

我们可以利用 DIG 命令很轻松的查询某一域名的 A、MX 等记录。如下图：

以 163.com 为例，查询对应的 A 记录：



同样的，如果想查询对应的 MX 记录，之需要将 a 改为 mx 即可。如下图：



最后的 A 记录是附加的信息，为了简化输出结果，我们可以将这部分内容过滤掉。

之需要配合适当参数即可。如下图：

同理，我们可以查询 PTR 记录、SOA 记录等等，这里不再演示。

这节仅仅是抛砖引玉，还有很多精彩的 DIG 应用未被列出，大家可以使用 dig -help 命令查询 DIG 更多的参数，由于本系列并非以 DIG 为主，所以这里就不再熬述其他内容，请见谅。

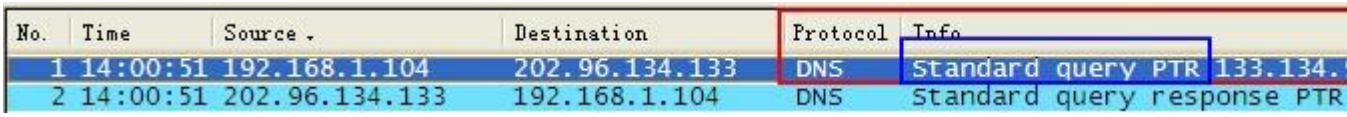
一起学 DNS 系列（十五）DNS 查询工具之 NSLOOKUP 的使用

上一节里我们讨论了有关 DIG 工具的用法，本节将对 windows 下 nslookup 工具的一些主要命令进行描述。

nslookup 是多数 win 系统中的一个内置命令，主要目的是用来测试域名解析，属于命令行工具，顺利工作的前提是已正常安装了 Tcp/Ip 协议簇。

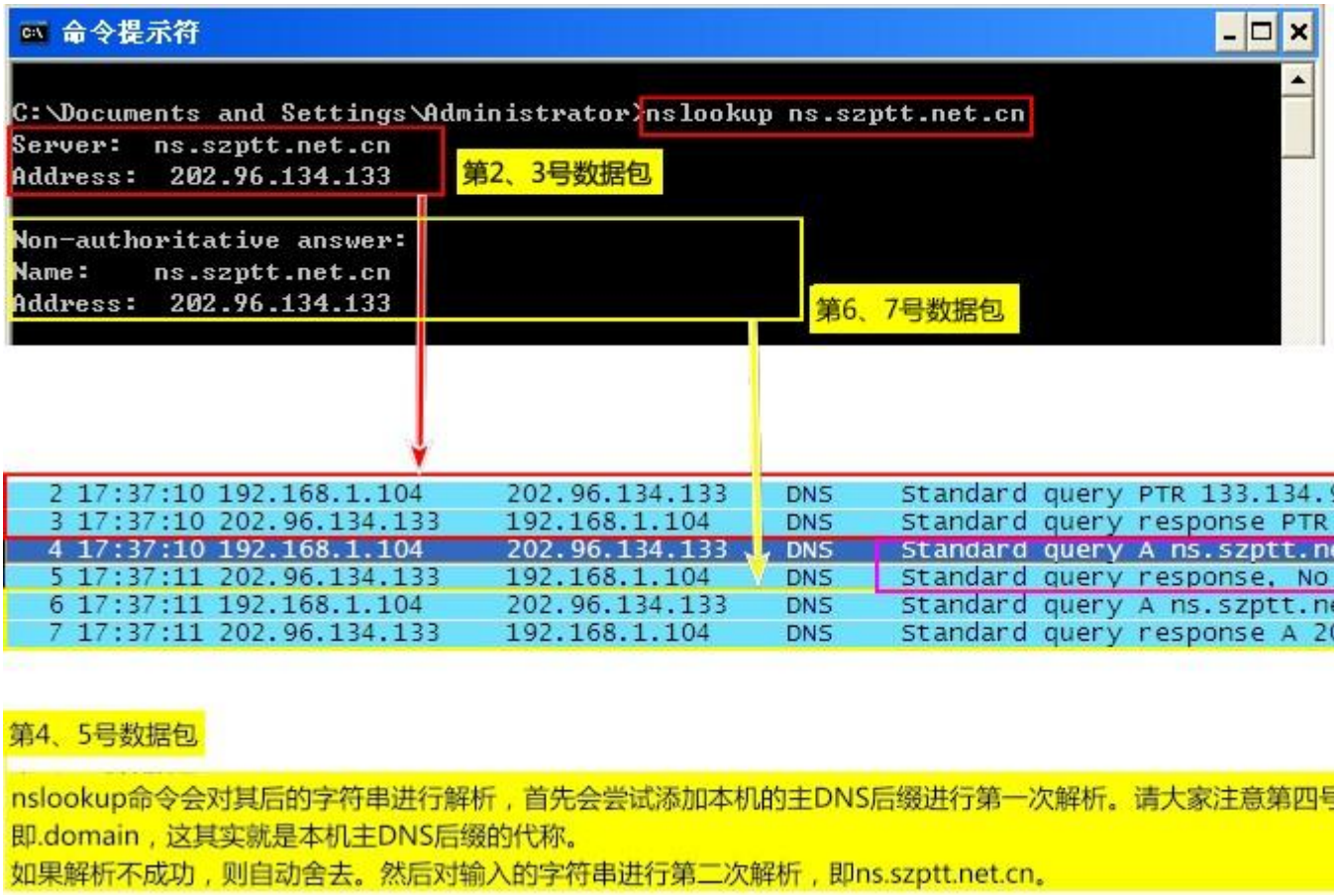
打开 CMD 的命令行界面，在不填写任何参数的情况下使用此命令，如下图：

如上图，输入 nslookup 后，返回了当前的 DNS 服务器的名称以及对应的 IP，其实，这一步骤的根本原理是对 202.96.134.133 这个 IP 的反向查询。如下图：



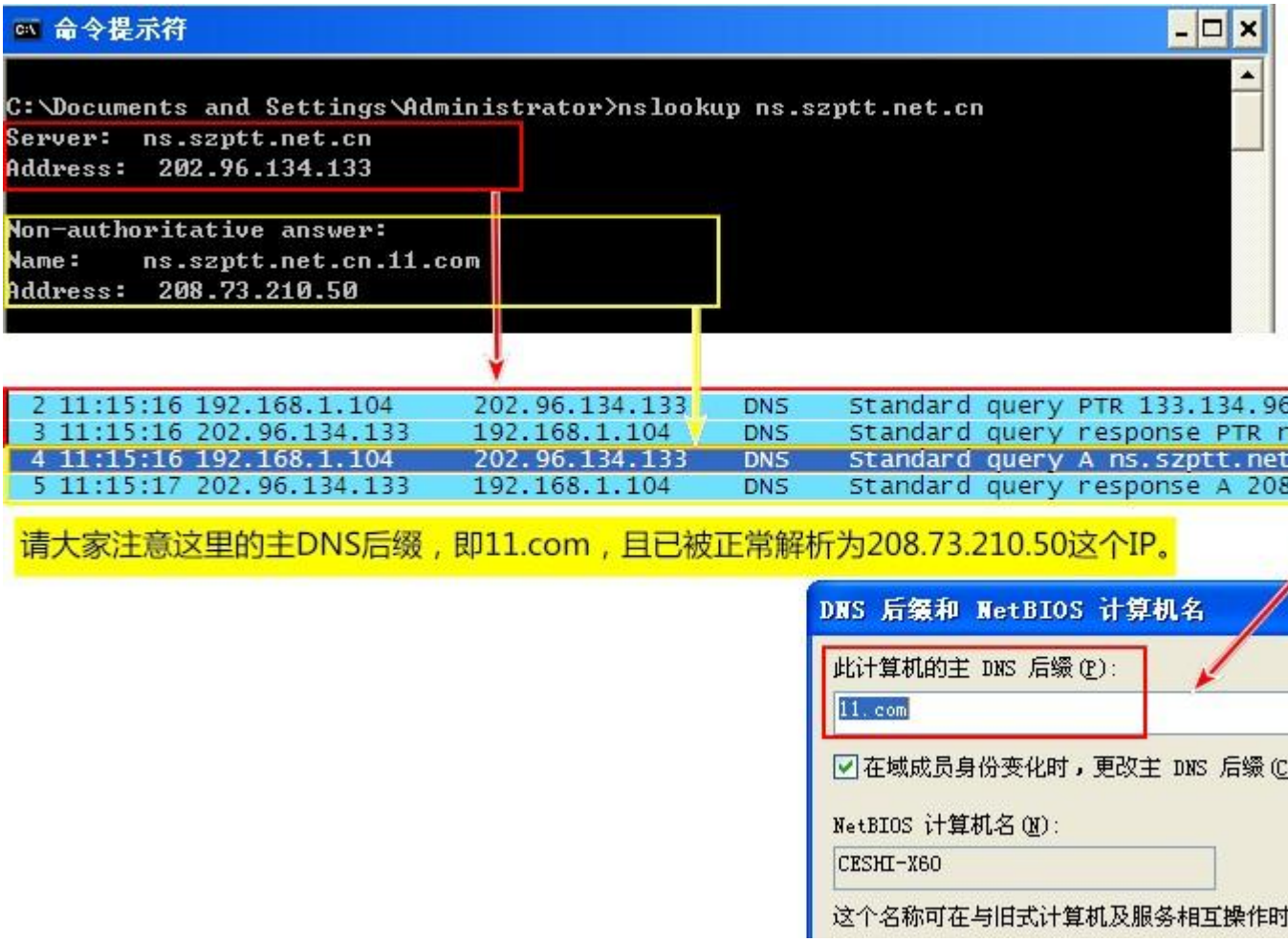
上图为与该操作同步产生的数据抓包图，实质是反向查询 DNS 的服务器名称。

当在 nslookup 后面加上名称或 IP 时，系统依然会执行一次反解，也就是重复上述过程，然后再对后面的内容进行解析。比如输入 nslookup ns.szptt.net.cn ，如下图：



为了便于分析，我把结果图和协议分析过程两者结合起来进行分析，大家可以清楚的看到，nsl
ookup 在解析字符串(也可能是域名)的整个过程。

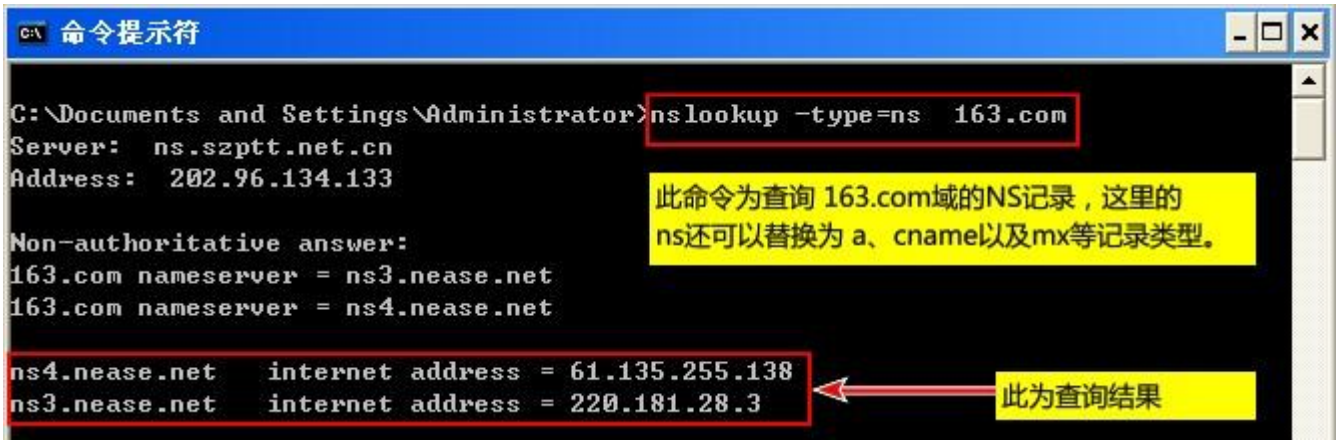
如上所述，如果我们设置一个主机的 DNS 后缀，如下图：



大家可以看到，.domain 被替换成 11.com，即设定的主机 DNS 后缀。

我们也利用 nslookup 命令解析 IP 对应的主机或域名名称。如下图：

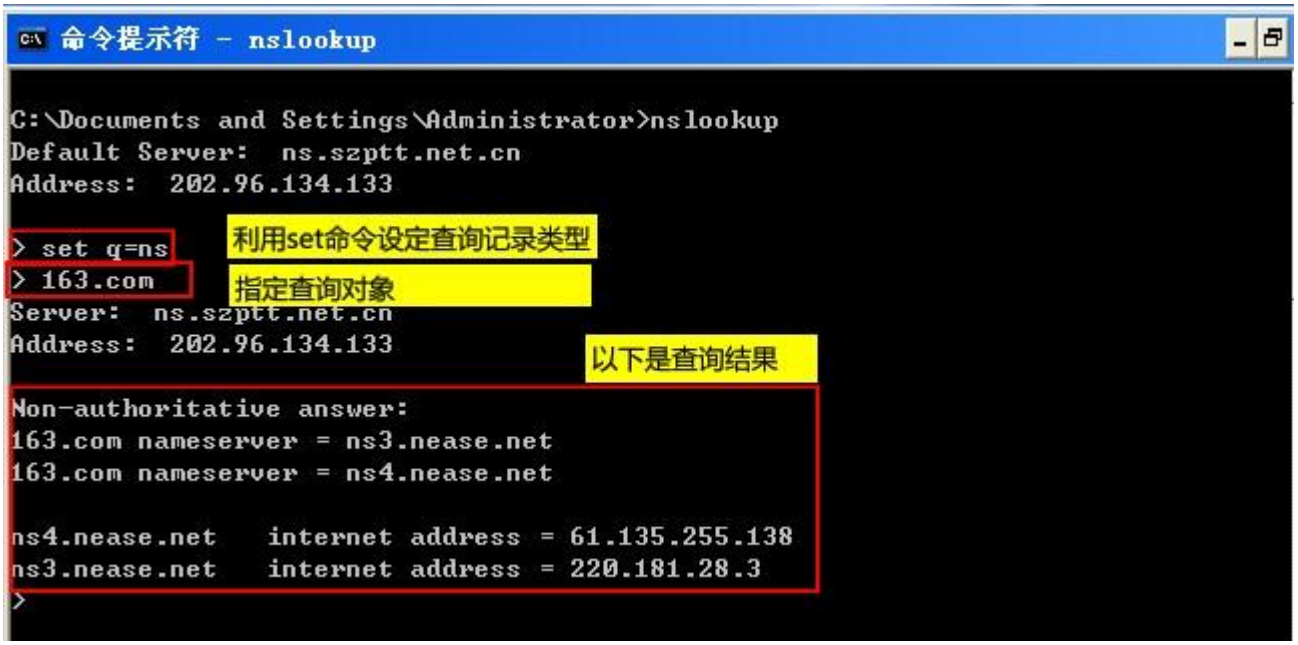
除此之外，还可以有其他的命令方式，如下图：



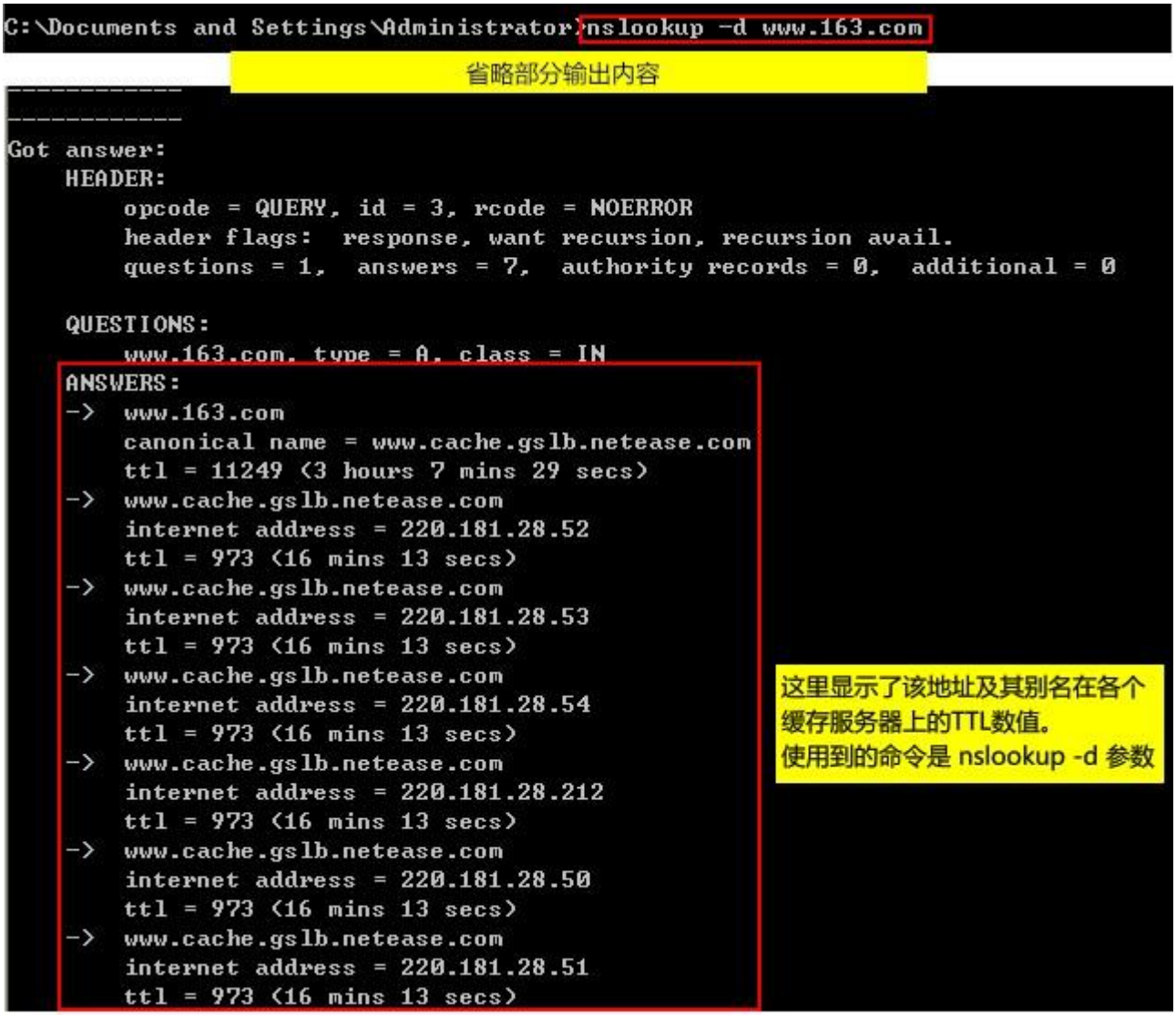
或者是

但要注意的是，在查询 cname 记录的时，后面需要一个完整的地址，而不是域名。

除此之外，还可以使用其他方式查询 A 记录或者其他域名记录。如下图：



当需要查询一个域名的 TTL 值时，我们就需要用到 -d 这个参数了，如下图：



上面所有的解析记录都来自默认的 DNS 服务器，即 ns.szptt.net.cn 上的**非权威回复**，关于何为非权威回复，在以前的章节中有描述，此处不多说。其实我们可以修改当前的响应服务器，也就是说可以利用 nslookup 的 server 命令来选定一个 NS 服务器来负责解析我们的请求，最近 google 公布了一组免费的 DNS 服务器，即 8.8.8.8 和 8.8.4.4，

我们就用第一个来测试一下，如下图：

第二个 DNS 测试结果雷同，只是 server 变成了 google-public-dns-b.google.com 而已。

这里只列举了一些比较常用的命令和查询方法，大家可以在 nslookup 的>提示符下输入？来查看更多的用法。

关于 nslookup 的用法就简单讨论到这里，谢谢大家！