A Typical NetInfo Setup

Alan M. Marcum

To make the conceptual explanations of NetInfo in this issue concrete, we'll use the example of a fictional NetInfo installation. Rhino Aviation, a typical company, manufactures accessories for the aviation industryÐthings like navigation radios, sun shades, and engine instruments. Rhino uses its NEXTSTEP computers the way many companies do: They have an application they've built that's critical to their businessÐa drafting programÐand they use applications that come with NEXTSTEP or that they've bought from third-party vendors.

Note: The document references in this and other articles in this issue refer to the books and articles listed in ªNEXTSTEP Networking References.º

RHINO'S COMPANY ORGANIZATION AND NETWORK DESIGN

Rhino Aviation has several departments: engineering, product marketing, sales, technical support, publicity, and information services. They have a connection to the Internet and two Class C network numbers; they couldn't get a Class B network number when they first applied and still can't get one, but they've outgrown a single Class C network.

Rhino's administrative information is organized into a three-level NetInfo hierarchy. Each department has its own domain, and there's a separate domain for the mail servers. A user can log into and use almost any

computer anywhere in the company as if it were on his or her own desk. The only computers that regular users can't access are:

· The firewall machine between Rhino's internal network and the Internet

· The mail servers

In addition to NetInfo, they use the Domain Name System (DNS, sometimes called BIND, **named**, or the resolver). If Rhino's network weren't connected to the Internet, they wouldn't use the DNS. Because it is, they use it both to resolve remote site name references and to resolve names internally. Rhino Aviation doesn't use the Network Information Service (NIS).

The domain hierarchy

Figure 1 shows part of Rhino's domain hierarchy. Each box represents a domain. The first line in the box is the domain name, the second line is the tag of the database used for the domain in this example, and the third line is the name of the computer that the domain's master server runs on. To the right of some domains are the names of computers that run the domains' clone servers. This is a three-level hierarchy; nine domains are shown.

Fig1_SimpleHeirarchy.eps ¬

Figure 1: *A portion of Rhino Aviation's domain hierarchy*

Who's serving whom

The **serves** properties in the root domain are set up like this:

name	Internet address	serves
super21	192.42.172.2	./Rhino
		mktg/network

mite	192.42.172.3	mktg/network
exec	192.42.172.4	./Rhino eng/network
mustang	192.42.172.5	./Rhino
cadet	192.42.172.6	eng/network
ovation	192.42.172.129	sales/network
sabre	192.42.172.66	ts/network ./Rhino
chaparral	192.42.172.98	info/network ./Rhino
mark21	192.42.173.130	mail/network
statesman	192.42.173.2	pub/network
pfm	192.42.173.67	
rhino	192.42.172.194	

(Some of these computers aren't shown in Figure 1.)

The **/eng** domain has these **serves** properties:

name	Internet address	serves
exec	192.42.172.4	./network exec/local ../Rhino
ranger	192.42.172.34	ranger/local
super21	192.42.172.2	../Rhino
cadet	192.42.172.6	cadet/local ./network

The **/mktg** domain contains these properties in the appropriate subdirectories of **/machines**:

name	Internet address	serves

mite	192.42.172.3	./network
		mite/local
super21	192.42.172.2	../Rhino
		super21/local
		./network
mustang	192.42.172.5	../Rhino

Each local domain has the following information, which happen to be the defaults, in **/machines**:

name	Internet address	serves
localhost	127.0.0.1	./local
broadcasthost	255.255.255.255	../network

Other first- and second-level domains

For simplicity, some second-level domains in the hierarchy aren't shown in Figure 1. There's nothing special or different about those not shown in the figure; they're left out just to make the diagram simpler. They're in Figure 2, as are several of the first-level domains. (Domains are numbered from the bottom upÐfor example, the local domain is first-level.)

Fig2_DomainHeirarchy.eps ¬

Figure 2: *Part of Rhino Aviation's domain hierarchy*

Two separate root domains are shown in Figure 2. The one whose master runs on **super21**, tagged **Rhino**, is for the entire companywide NetInfo hierarchy; the one whose master runs on **rhino**, tagged **local**, is for the firewall machine, called **rhino** (or **rhino.Rhino.com**). In other words, the firewall machine *isn't* bound to the rest of the NetInfo domain hierarchy. For clarity, we'll call the ªrealº root domainÐthe top of the company-wide

domain, tagged **Rhino**Ðthe ªroot domain,º and call the firewall's domain the ªfirewall domain.º

This sort of domain hierarchy configuration is pretty common. It maps closely to the recommendations in the *NEXTSTEP Network and System Administration* manual, the NEXTSTEP System Administration classes, and past *support bulletin* articles including Tucker 1992 and Cottle, ªWhen One Server Isn't Enough,º 1993.

The information in **/users** in the firewall's domain is separate from that in the root domain. This is critically important for strong network securityÐfor example, it forces someone trying to break into your network to break into two separate accounts. For further explanation, see Tucker 1992 and Garfinkel and Spafford 1991.

As noted especially in Cottle, ªWhen One Server Isn't Enough,º 1993, the aliases database doesn't reside in the root domain, but rather in the **/mail** domain. This decreases the size of the root domain dramatically, without decreasing functionality. (The alias information is needed only by the mail servers anyway.) Typically, the root domain has more clones than other domains, especially mail-specific domains such as **/mail**. Decreasing the size of the root domain allows Rhino Aviation's network to grow much larger, more easily.

BALANCE IN A NETINFO HIERARCHY

In other NetInfo configuration examples you've seen elsewhere, you might have seen a ªbalanced hierarchy.º With a balanced hierarchy, every local domain has the same number of ancestors. The Rhino Aviation hierarchy has a nearly balanced hierarchy. However, balance isn't required.

How might the hierarchy not be balanced? Assume that most of the

network uses a three-level domain hierarchy. If some local domain's parent is the root domain, then that portion of the network uses only a two-level domain hierarchy. Similarly, if some second-level domain's parent *isn't* the root domain, then that portion of the network uses a four-level or deeper hierarchy.

You might use an unbalanced hierarchy if one second-level domain has become very large and needs to be divided. You might also use one if one second-level domain represents a department that has grown and now has ªsubdepartments.º Or, using an unbalanced hierarchy might just make sense. As an example, look at Figure 3.

Fig3_UnbalancedHeirarchy.eps ¬

Figure 3: *An unbalanced domain hierarchy*

In this example, all domains except for **/sales** are the same as in the previous illustration. The **/sales** domain hierarchy is shown all the way to the local domains. In this case, when referring to the domain hierarchy from, for example, **genesee**, the **network** domain (**/sales/syracuse**) is the second-level domain, the **Department** domain (**/sales**) is the third-level domain, and the root domain is the *fourth-level* domain.

This third example is just for illustration. For the rest of the issue, you can assume we're referring to the hierarchy in Figure 2.

WHERE TO NOW?

With this groundwork laid, the remaining articles in this issue examine how things stand before NetInfo starts, how servers get information they need about NetInfo through the lookup server, what binding and connecting are and how they work, how changes to a domain are propagated, and how to handle particularly tricky troubleshooting.