Searching NetInfo in sendmail Config Files

Alan M. Marcum

You can use the version of sendmail in NEXTSTEP Release 3.3 to look up information in NetInfo configuration files. This makes it easier to replace login names with more recognizable e-mail aliases in mail.

Translating for clarity

To provide for some aspects of address rewriting, **sendmail** can look up information in an NIS mapÐthat is, look up a keyÐand substitute other information found in that mapÐsubstitute
a value. You can access this feature through the **sendmail** configuration file, using the **${Z¼$}** directive.

One of the most frequent uses of this NIS lookup mechanism is to translate the login name of a user sending mail into some sort of canonical address for that user. For example, Sandy Kabir's login name might be **sandy**, or **skabir**, or **sandyk**, or **sk**, or **eng672**. However, Sandy might

well have an e-mail alias, though, of **Sandy_Kabir**Ðsomething with universally recognizable syntax, semantics, content, and meaning. If Sandy's login name is translated to a canonical alias through inverse alias translation, the recipient of any of Sandy's messages will much more easily recognize the sender, and others will find it much easier to figure out how to reach Sandy
by e-mail.

Until recently there was no way to look up something in the NetInfo databases. This was a minor sore point with many NEXTSTEP site administrators, who wanted to implement inverse alias translation but didn't want to run NIS. (And golly, who could blame them?)

Late in 1993, NeXT released a patched version of **sendmail** (version 5.67e) to address some significant security concerns. At the same time, NeXT quietly slipped a new feature in this version of **sendmail**: the ability to perform NetInfo lookups. Now, NEXTSTEP Release 3.3 includes this newer version of **sendmail**, so the new feature is now available to all.

Defining NetInfo ªMapsº

To access NetInfo from a **sendmail** configuration file, you first need to define where the data should reside in NetInfoÐthe NetInfo ªmap.º Specifically, this map includes the domain or database, and the directory. In the case of the domain or database, you can use the path of the domain to search (for example, **/** or **/mail**), or the address of the database, using tagged domain notation (such as **sabre/network** or **cadet/Rhino**). Specifying a domain allows any of the servers of that domain to respond to the request, whereas specifying a database forces a particular server to be the responder (for example, **netinfod** Rhino on cadet). The advantage of specifying a domain is reliability; the advantage of specifying a database is performance. You can omit the domain or database entirely to specify a search of the NetInfo hierarchy; this provides the maximum flexibility, but with a potential performance penalty associated with accessing multiple NetInfo domains.

One thing to note: NeXT supports this feature only minimally.

In many cases, the computer providing mail services runs a copy of the NetInfo database for the domain you need to search. If so, then the additional reliability of specifying a domain is irrelevant: You can access the database on the local computer (such as **localhost/Rhino**), and you'll know that if the mail server is up, running, and processing mail, the NetInfo server on that computer is likely to be up and responding to requests too.

Here's the syntax for defining the NetInfo domain or database and directory:

Kmapname **netinfo [**domain:**]**dir prop_key

mapname is a single-letter name for this mapping, domain is the domain or database, dir is the NetInfo directory to be searched, and prop_key is the key of the property whose first value should be substituted. Two examples:

```
KU    netinfolocalhost/Rhino:/usersemail
Kh    netinfo/machinesip_address
```

In the first example, the **U** map refers to **/users** in the NetInfo database tagged **Rhino** running on the local computer. When this map is used to translate a token, the local database Rhino is referenced; the first subdirectory in **/users** whose name property matches the token is read, and the first value of the **email** property is substituted for the token. (Note that of the NEXTSTEP system administrator applications, only NetInfoManager provides access to such a property.) If **sk** is Sandy Kabir's login name, and **Sandy_Kabir** is the appropriate canonical alias, then the search will find the NetInfo directory **/users/sk** and substitute **Sandy_Kabir**, the first value of the **email** property, for the token **sk**.

In the second example, the **h** map refers to the **/machines** directory throughout the domain hierarchy. When this map is used, the first value of the **ip_address** property in the matching subdirectory is substituted for the token.

To perform the lookup, you use the directive **$(¼$)**; for example, **$(U$1$)**. Details on using this directive are provided below.

How might the **U** map defined above be used to provide inverse alias translation, so that mail comes from **Sandy_Kabir@Rhino.COM** instead of **sk@Rhino.COM**? First, let's see how it has been done using NIS, prior to **sendmail** version 5.67e.

In the Old Days: NIS Lookups

To invoke NIS lookups, you would first define the name of the NIS map to be searched. The name is assigned to a macro as its value. For example:

```
DZmail.byaddr
```

Then you'd perform the lookup in the rewrite portion, or right side, of a rule. (This is the second clause in the rule; the first clause is the match portion, or the left side.) The syntax is:

```
${directive$}
```

The directive has two parts. The first part is the letter of the macro that contains the name of the NIS map to be searched. The second part is the string on which the search should be performed.

For example, the following right side excerpt will look for the third token (**$3**) in the **mail.byaddr** map, assuming that the **Z** macro's value is **mail.byaddr**:

```
${Z$3$}
```

For a concrete example, Figure 1 shows Ruleset 22 from **sendmail.mailhost.cf**. Ruleset 22 is used to rewrite sender addresses for mail going out to the Internet. (**Boldface** highlights the particularly relevant bits.)

```
DZmail.byaddr

S22
R$*<@LOCAL>$*$:$1
R$-<@$->$:$>3${Z$1@$2$}invert aliases
R$*<@$+.$*>$*$@$1<@$2.$3>$4already ok
R$+<@$+>$*$@$1<@$2.$m>$3tack on our domain
R$+  $@$1<@$j>tack on our hostname
```

Figure 1: *A map containing the inverse of mail.aliases*

The second rewrite rule–the one with the comment ªinvert aliasesºÐis where the translation is done. This rule matches one or more tokens, followed by **<@**, followed by one or more tokens, followed by **>**. (The angle-brackets,**<** and **>**, are added internally by **sendmail** to assist in its rewriting bookkeeping.)

So, this rule will match addresses like the following (brackets removed for clarity):

```
sk@rhino
amm@sabre
root@cadet
```

In the highlighted portion of the rule (**${Z$1@2}**), an NIS lookup in the map **mail.byaddr** is performed. Where did **mail.byaddr** come from? It's the value of the **Z** macro, which was defined just before Ruleset 22 is defined. **sendmail** searches for the first token, followed by **@**, followed by the second token. If **sendmail** finds this string, it substitutes the NIS value for that key in place of the key. In each of the examples above, the entire portion of the address before the @ is
the first token, and the rest of the address after the @ is the second token.

Here's a further example. Assume that the **mail.byaddr** map contains the following:

```
amm@sabre: Alan_Marcum
root@cadet: SuperUser
sk@rhino: Sandy_Kabir
```

If Ruleset 22 is invoked on each of the addresses from the examples, it produces the results in Figure 2.

Before	After
sk@rhino	Sandy_Kabir
sk@sabre	sk@sabre
amm@sabre	Alan_Marcum
root@cadet	SuperUser

Figure 2: *Invoking Ruleset 22*

Note that **sk@sabre** isn't changed, because **sk@sabre** isn't in the databaseÐonly **sk@rhino** is.

Converting to NetInfo

Now let's convert Ruleset 22 to use NetInfo. First, the way the map is defined is changed, as described above. Let's assume that we'll add the property **email** to each user's account record in NetInfo, and that this property will have as its value the preferred alias for that user. For example, here's the output from **niutil -read** showing **sk**'s new user record:

```
name: sk
passwd: SZ0QqjPwZdavo
uid: 726
gid: 20
realname: Sandy R. Kabir
home: /Net/rhino1/Users/sk
shell: /bin/csh
_writers_passwd: sk
email: Sandy_Kabir
```

Case is important in defining the tag of the property: **EMail** is different from **email**.

Assume that this user record is in the root domain, a server for that domain is running on the computer providing mail service, and the root domain's database on this computer is tagged **Rhino**. Also, **Sandy_Kabir** must be a mail alias for **sk**.

Then, we define a mapÐlet's use the name **Z**, for consistency with our NIS-based example. See the first line in Figure 3.

```
KZnetinfo localhost/Rhino:/usersemail

R$-<@$->$:$>3$(Z$1$)@$2$invert aliases

R$-$:$>3$(Z$1$)inverse alias translation
```

Figure 3: *Steps for changing rules and defining a map*

Now, we modify the second rule in Ruleset 22 to read as the second line in Figure 3.

We've made two changes. First, we've changed the **${¼$}** directive to **$(¼$)**. Though the former will still work, the latter is the preferred syntaxÐit's the syntax used in **sendmail** version 8. Second, we're only searching on the first token, rather than the first token followed by @ followed by the second token. This way we can use the **/users** directory from NetInfo with minimal modification.

You can make this a little nicer for your users by applying inverse alias translation to all messages, not just those destined for the Internet. To do this, remove that second rule from Ruleset 22 and add the third line in Figure 3 to Ruleset 1.

Now, **sendmail** applies inverse alias translation to the sender addresses of all messages that originate locally, not just those destined for the Internet. This is because Ruleset 1 is applied to all sender addresses, prior to applying the mailer-specific sender rewriting rule, such as
Ruleset 22.

In Summary

Sometimes little details make a noticeable difference. You've been asking for it, now you've got it: the ability to do NetInfo lookups in **sendmail** configuration files. There are other uses for the feature in addition to inverse alias translation, such as directing messages to explicit Internet addresses; such uses are, as the saying goes, ªleft as an exercise for the interested reader.º

Oh, and incidentally: If you want more information about this facility, including both some additional features of NetInfo lookup and a description of the map lookup capability, see the **sendmail** appendix in the *NEXTSTEP Network and System Administration* book for Release 3.3.

Alan M. Marcum is a member of the Premium Support Team and specializes in large networks.