Symbolics IP/TCP Software Package

Overview of IP/TCP Software Package

The IP/TCP software package enables the Symbolics computer to communicate with other systems supporting the ARPA Internet protocol family. You can locate these systems on either the local Ethernet or another Internet network connected to the local Ethernet through a gateway. The Internet protocols supported include:

- Remote login (TELNET and SUPDUP)
- File transfer (FTP and TFTP)
- Electronic mail and messages (SMTP)

These services are accessible to the Symbolics computer user through the generic network system. See the section "Symbolics Generic Network System".

You can also add support for new or special-purpose Internet protocols to the basic system. See the section "Defining a New Network Service".

For references to detailed information about the Internet protocol family: See the section "References to IP/TCP Protocol Specifications".

IP/TCP Installation

This section contains instructions for installing the IP/TCP system and configuring the site for IP/TCP. For further information on IP/TCP: See the section "Reference Information on IP/TCP".

Overview of IP/TCP Installation

The following sections describe the process of installing IP/TCP at your site. You must have completely installed Genera before beginning the installation of IP/TCP.

The installation contains the following steps:

1. Restoring IP/TCP from tape: *All sites*
2. Configuring your site for IP/TCP: *New sites only*
3. Loading the IP/TCP system on one machine and saving the world: *All sites*
4. Copying that world to other machines at your site: *All sites that own an IP/TCP Site License*
5. Installing the Internet Domain Names system: *Optional*

You should also consider the security of your IP/TCP network. For further information on controlling the use of servers (such as TELNET and FILE) with access to sensitive information: See the section "IP/TCP Security Considerations".

Restoring IP/TCP from the Distribution Tape: *All sites*

This step restores the IP/TCP software from the distribution tape to the SYS:IP-TCP; directory.

1. Place the distribution tape in the cartridge tape drive of any Symbolics computer at the site.

2. Load the distribution tape by entering the following command to a Lisp Listener:

   ```
   Restore Distribution
   ```

   The distribution loader prompts you as follows:

   ```
    ** A tape is needed to read distribution
   Enter a tape spec [default Local: Cart]:
   ```

   If the tape is loaded on the local machine, press RETURN. If the tape is not loaded on the local machine, you can press HELP for a list of choices. Everything on the tape is restored.

3. Remove the tape from the cartridge tape drive.

4. Cold boot the machine.

Configuring the Site: *New Sites Only*

Assigning Internet Addresses: Optional for New Sites Only

For background information on network addressing: See the section "Network Addressing".

After installing IP/TCP from the distribution tape, you can assign Internet addresses on the appropriate networks to each host. This is a paper and pencil step that does not involve the computer. Later you enter the Internet address in the address attribute of the host object for each host that will use IP/TCP protocols.

The requirements for Internet addresses are:

- Each host that will use IP/TCP needs a unique Internet address in the correct format. See the section "Format of Internet Addresses".

- Gateway or multihomed hosts should have addresses assigned for each network to which they are connected.

Symbolics has some suggestions for assigning Internet addresses. These are not technical requirements.

- Obtain a valid ARPA Internet network number from SRI-NIC.

  You can use IP/TCP whether or not your site is connected to the ARPA Internet. However, even if your site is not now connected to the ARPA Internet, we strongly suggest that you obtain a valid Internet network number for your local Ethernet network. By using a valid Internet network number, you can set up your site now so that at any future time you can easily connect to the ARPA Internet without having to reassign Internet addresses for every host at your site.

  To obtain an Internet address: See the section "How to Obtain an Internet Address". The Internet address you receive is the network part of the address. You assign the host number part of the address yourself.

- Derive a host's Chaos address from its Internet address.

  Hosts that support both the Chaos and the IP/TCP protocols have two network addresses: a Chaos address and an Internet address. We recommend an addressing scheme that maps an Internet address into a Chaos address. This scheme makes sense for all sites, whether or not they are connected to the ARPA Internet. For more information: See the section "Mapping an Internet Address into a Chaos Address".

Configuring non-ARPA Internet IP/TCP Sites: *New Sites Only*

This section is intended for new sites that are not connected to the ARPA Internet. For new sites that are connected to the ARPA Internet: See the section "Configuring the Internet IP/TCP Sites: New sites only".

You need to enter the namespace editor, create a network object of type Internet, and save that object.

Enter the namespace editor by selecting it from the System menu or by giving the Edit Namespace Object command. Click on [Create Object]. The type of the object is [Network]. The name and type of the new network object is Internet. Also enter the name of your site to the site attribute.

The final record looks similar to this:

```
Nickname: Name
Site: YOUR-SITE
Type*: INTERNET
Subnet: a pair of a token and zero or more pairs of a global name and a token
User Property: a user property pair of a global name and a token
```

Click on [Save Object] to save the new object.

Configuring the Internet IP/TCP Sites: New sites only

This section is intended for new sites that are connected to the ARPA Internet. For new sites that are not connected to the ARPA Internet: See the section "Configuring non-ARPA Internet IP/TCP Sites: *New Sites Only*".

Describing the process of physically configuring the Internet gateways for your network is beyond the scope of this document. We assume that your Internet includes one or more Ethernet networks that have been assigned Internet network numbers and that are connected to the rest of the ARPA Internet through one or more gateway machines.

The steps in this section must be performed on the machine that is the primary namespace server at your site. The installation will fail if you perform these steps on a machine other than the primary namespace server.

1. Cold boot the system.

2. You should already have determined the Internet addresses to be assigned to the namespace server and to its primary gateway to the rest of the Internet. Do the following:

```
(si:login-to-sys-host)
Load System Ip-tcp
```

   The Load System command might print warning messages regarding changed flavors; you can ignore these. If you want to use the internet namespace, evaluate this form:

```
(tcp:initialize-internet-namespace)
```

   This form starts the following dialogue. If you answer any of the questions below by pressing RETURN, it is the same as choosing the default answer. The defaults are shown in italics in the dialogue.

```
Name of the namespace to create (INTERNET)? INTERNET
        default is INTERNET
Directory where INTERNET namespace files should be kept
        (default LOCAL:>sys>site>)? LOCAL:>sys>site>
Internet address of your local host?
        default is: address on the Ethernet of the namespace server
Name of your internet gateway?
        default is: name of the gateway on the Ethernet
Internet address of gateway?
        default is: address of the gateway on the Ethernet
```

The function **tcp:initialize-internet-namespace** initializes a prototype namespace object for the Internet and makes the local host its primary namespace server. It also adds the new namespace to the site's local namespace's search list and adds an address for the local host on the new network.

3. Use the Save World command to save the current world load. This world should be used by the namespace server from now on, but we suggest that you do not copy this world to other machines. If you are not using the Internet Namespace, skip to step five. See the section "Save World Command".

4. Install the NIC host table information in the namespace database. Follow this example:

```
(tcp:install-nic-host-table)
Output file (default LOCAL:>sys>site>internet-hosts.text)?
        LOCAL:>sys>site>internet-hosts.text
```

The function **tcp:install-nic-host-table** reads the NIC host table and loads it into the INTERNET namespace object file. This step can take quite a long time because it involves a file transfer of a long file from a host at SRI.

The namespace itself is updated by the function **neti:read-object-file-and-update**:

```
(neti:read-object-file-and-update "internet" :host)
```

You can repeat this step at any later time, to install a more recent version of the NIC host table.

5. Set up Internet-Domain-Name attribute.

New IP/TCP sites that are connected to the ARPA Internet must enter a value for the Internet-Domain-Name attribute of the namespace object that represents the local namespace. For example, the SCRC namespace has the Internet-Domain-Name attribute set to "SCRC.Symbolics.COM". See the section "Internet Domain Name Namespace Attribute".

Updating the Namespace for IP/TCP: *New Sites Only*

If you are unfamiliar with the namespace system: See the section "Setting Up and Maintaining the Namespace Database".

In this step you edit the host object for each host that will use IP/TCP protocols. If the host object does not already exist, you should create it. If you need to create host objects, enter the System type and Machine type attributes for the hosts. Probably most host objects for Symbolics computers already exist, from steps performed in the Genera installation. However, it is important to create (or edit) the host object for every host (not just Symbolics computers) on the network that uses IP/TCP protocols.

Each host object needs values for the following attributes:

system-type The operating system of the host. Examples: `LISPM` (for any release of Symbolics software), `UNIX` (for UNIX versions prior to 4.2), `UNIX42` (for UNIX version 4.2 and later), `VMS4.4` (for VMS versions 4.4 and later).

machine-type The hardware type of the host. Examples: `3600` (for any Symbolics 3600-family computer), `vax`.

address A unique Internet address. Example: `INTERNET 192.10.41.48`.

service A service attribute is necessary for each IP/TCP service the host supports. The complete list of services appears later in this section.

server-machine For server machines only (such as the site's file server), the value of this attribute should be `"YES"`.

From a Symbolics computer, choose [Namespace Editor] from the System menu or give the Edit Namespace Object command to begin editing the namespace database. When the namespace editor prompts you for the type of object to edit, type Host, and then enter the name of any host at the site that will use IP/TCP protocols. Then the namespace editor prompts you for the name of the namespace. Type the name of the namespace and press `RETURN`.

1. For newly created hosts, enter values for the **System-Type** and **Machine-Type** attributes.

   This following example shows the attributes for a Symbolics 3640 computer host:

   ```
   System Type*: LISPM
   Machine Type: 3600
   ```

2. Enter the Internet address.

   Add the Internet network address of the host. Be sure to add all addresses for any gateway and multihomed systems.

   ```
   Address: INTERNET 192.10.41.48
   ```

3. Enter the appropriate IP/TCP **service** attributes.

   A *server machine* is able to provide some type of service to other hosts. For example, a file server is a computer given the task of storing files for a large number of users at a site.

   Add the following **service** attributes to the namespace host object for each Symbolics computer that should provide the IP/TCP services:

```
Service: MAIL-TO-USER TCP SMTP
Service: STORE-AND-FORWARD-MAIL TCP SMTP
Service: EXPAND-MAIL-RECIPIENT TCP SMTP
Service: CONFIGURATION TCP CONFIGURATION
Service: FILE TCP TCP-FTP
Service: FILE TCP NFILE
Service: FILE UDP TFTP
Service: LOGIN TCP TELNET
Service: LOGIN TCP SUPDUP
Service: LOGIN TCP 3600-LOGIN
Service: SHOW-USERS TCP ASCII-NAME
Service: SEND TCP SMTP
Service: TIME UDP TIME-SIMPLE-MSB
Service: LISPM-FINGER UDP LISPM-FINGER
```

Note that the STORE-AND-FORWARD-MAIL, MAIL-TO-USER, and EXPAND-MAIL-RECIPIENT services are available only if your site is running the store-and-forward mailer provided by Symbolics. See the section "Symbolics Store-and-Forward Mailer".

It is important to add the appropriate **service** attributes for the non-Symbolics computers at the site. You should consult the documentation for other systems' IP/TCP packages to determine which servers they supply. Add the appropriate **service** attributes to those host objects. The ASCII-NAME service is sometimes called FINGER in the documentation provided by other companies; however, it should be entered into the **service** attribute as ASCII-NAME.

4. Change the value of the **Server-Machine** attribute to be YES.

   For any Symbolics computer that is to be a server machine, such as a file server, edit the attribute **Server-Machine** in the host object. Change the value to be YES. The resulting entry is:

   ```
   Server Machine:  YES
   ```

5. Save the host object.

   Click on the [Save Object] command to save the host object you have edited.

6. Edit and save the remaining host objects.

   Click on [Edit Object] to edit another host object. Go through the steps in this section for each host at the site that uses IP/TCP.

Loading IP/TCP on a Symbolics computer: *All sites*

It is important to load IP/TCP on the namespace server, immediately save the world, and use that world for the namespace server from now on. (Note that new ARPA Internet sites have already created a world containing IP/TCP for the namespace server to run.)

You can also load IP/TCP on any Symbolics computer that will use IP/TCP. If you have a Site License, and are going to copy a world containing IP/TCP from one machine to another (and thus distribute it throughout the site), we recommend loading IP/TCP on a Symbolics computer that is not the namespace server, and saving and distributing that world.

To load IP/TCP on an individual Symbolics computer, type the following command at a Lisp Listener:

```
Load System ip-tcp
```

IP/TCP is now loaded.

We recommend that you use the Save World command and use this world from now on. See the section "Save World Command".

If you do not immediately save the world, you must give the following command before using IP/TCP:

```
Enable Network INTERNET
```

Distributing the IP/TCP World: *All sites with a Site License*

Once you have loaded IP/TCP on one Symbolics computer and saved the world, you can share that world with the other hosts at your site by using the Copy World command. We recommend that you do not copy the world from the primary namespace server to other machines at the site, but instead load IP/TCP on another machine, and save and copy that world.

Alternatively, each Symbolics computer can load the IP/TCP system individually and save the world. See the section "Loading IP/TCP on a Symbolics computer: *All sites*".

Registering Internet Gateways in the Namespace

You can enable Symbolics computers on different Internet network/subnets to communicate by specifying an Internet gateway host. In order to create an internet gateway host that the machines on both networks recognize, you must create a host object in the namespace database containing:

- The Internet addresses assigned for each network to which the gateway connects.
- A service entry for Gateway service.

For example, you must add the addresses of each network to the host object in the namespace database:

```
Address: INTERNET 128.81.38.8
Address: INTERNET 128.81.73.1
```

Additionally, you must add a service entry to the host object in the namespace database:

```
Service:  GATEWAY IP INTERNET-GATEWAY
```

The `GATEWAY IP INTERNET-GATEWAY` service entry enables a host to forward IP packets. You have to specify this service in the namespace object of the gateway machine, save the object, and reset the network.

Note that you can use the `DEFAULT-INTERNET-GATEWAY` service for forwarding IP packets to another gateway when no route using the `GATEWAY IP INTERNET-GATEWAY` service is available.

After you add the network addresses and service entry to the host object in the namespace database, the gateway host appears in the Peek window under Interfaces in the network frame of the local Symbolics computer as one of the following:

- *Gateway-host-name Gateway-internet-address* Alive
- *Gateway-host-name Gateway-internet-address* Dead

If the gateway appears in the Peek window as `Alive`, the gateway is responding to the local host. You can now connect the local host to the machines on the other internet network/subnet using the gateway.

If the gateway appears in the Peek window as `Dead`, the gateway is not responding to the ICMP request required by the IP protocol. You can verify whether the gateway is responding by evaluating this form:

(**tcp:send-icmp-echo** *"gateway-host"*)

For more information, see the function **tcp:send-icmp-echo**.

The `DEFAULT-INTERNET-GATEWAY` host User-Property enables a host to forward IP packets to another gateway when no route using the `GATEWAY IP INTERNET-GATEWAY` service is available. It is only used by the lisp machine gateway routing code.

Installing the Internet Domain Names System: *Optional*

This section contains the installation procedure itself, and some background information which might be useful. For additional information, see the section "Internet Domain Names".

Installation Procedure for Internet Domain Names

Any site that decides to use the Internet Domain Names capability should perform these steps.

1. Become familiar with the issues of Internet Domain Names: *Recommended*

   The Symbolics documentation introduces the concepts and describes how Genera uses Internet Domain Names. See the section "Internet Domain Names".

   We strongly recommend that you obtain and read the documentation available from SRI-NIC that fully describes the concepts underlying Internet Domain Names. See the section "References to IP/TCP Protocol Specifications".

2. Assign a Central Name Resolver: *Optional*

   Decide if you want to assign a host at the site to be a central name resolver. If so, you should add the following service attribute to the host object of the central name resolver host:

   ```
   Service: Set: DOMAIN CHAOS DOMAIN
   ```

   Sites that support IP/TCP should also enter:

   ```
   Service: Set: DOMAIN TCP DOMAIN
   ```

3. Enter the addresses of the Root Domain Servers: *Required for external queries.*

   It is important that the hosts at the site know the addresses of the root Name servers. This information should be stored in the **root domain server address** attribute of the site object. When these addresses are entered, it makes it possible for hosts at the site to seek name information outside of the site, using the standard Domain Names protocols. If the **root domain server address** attribute is left empty, the hosts at the site cannot make external queries for information on hosts outside the site.

   For IP/TCP sites that are connected to the ARPA Internet, the site object should contain the addresses of the root domain server hosts. (Note that these are the current root domain servers, and this list changes periodically. To get the most current root domain server addresses, see the following file: `sri-nic:netinfo:<netinfo>root-servers.txt`.) Here are the attributes:

Hostname	Net Addresses	Server Program
NIC.DDN.MIL	10.0.0.51 26.0.0.73	JEEVES (Tops20)
A.ISI.EDU	26.3.0.103 128.9.0.107	JEEVES (Tops20)
AOS.BRL.MIL	192.5.25.82 128.20.1.2	BIND (UNIX)
GUNTER-ADAM.AF.MIL	26.1.0.13	JEEVES (Tops20)
C.NYSER.NET	192.33.4.12	BIND (UNIX)
TERP.UMD.EDU	128.8.10.90	BIND (UNIX)
NS.NASA.GOV	128.102.16.10	BIND (UNIX)

Any sites that are not connected to the ARPA Internet, but are using Internet Domain Names should enter the network and network address of the root Name servers you are using.

4. Enter the Internet Domain Name for the local namespace: *Required*

   In this step you specify the Internet Domain Name to be associated with the namespace in which local hosts are registered, by editing the **Internet Domain Name** attribute of the namespace object that represents the local namespace itself. For further information: See the section "Internet Domain Name Namespace Attribute".

   For example, the SCRC namespace object might have this attribute:

   ```
   Internet Domain Name:  SCRC.Symbolics.COM
   ```

5. Install the Name server software: *Optional*

   Now decide if your site is going to provide a Names server host to provide the ARPA Internet community with information about the hosts in your domain. Often it is convenient to use a single Symbolics computer as the central name resolver (which serves hosts within your site) and as a Name server (which serves hosts outside of your site); however, this is not necessary. These two functions can be performed by separate hosts if that is more convenient.

   The software that enables a Symbolics computer to act as a Name server is a separately loadable system. It is distributed on one of the system tapes. Load the system:

   ```
   Load System Domain-Name-Server
   ```

   Save the world containing the Name server software. The server machine should use that world from now on.

6. Edit the launch file for your site: *Optional*

Note that the Genera implementation enables the Name server to use the information already stored in the namespace database. When services are enabled on the Name server host, it looks for the file named SYS:SITE;LAUNCH-DOMAIN-SERVER.TEXT, which indicates where the data files that contain all the name information are stored. If that file does not exist or is empty, the server uses only the namespace database files. For many sites, the launch file is unnecessary.

The server can also use the information stored in Dialnet registries, if you have an entry in the launch file that indicates to the server where the Dialnet registries are stored. A sample launch file is provided on the tape, with the entries commented out. You can edit that file for your site if you want to make Dialnet or other names data files available to the server.

Here is a sample launch file:

```
;       Sample boot file for Name server
;
; type        domain                    source file or host
;
domain     yoyodyne.com
primary    yoyodyne.com           sys:site;yoyodyne-domain.text
dialnet    dialnet.yoyodyne.com   sys:site;private-dialnet-registry.lisp
dialnet    dialnet.symbolics.com  sys:site;public-dialnet-registry.lisp
```

For details on the format of the launch file, see the section "The Launch File". In this case, the data file is SYS:SITE;YOYDYNE-DOMAIN.TEXT. For details on the format of data files, see the section "The Internet Domain File".

Registering Your Domain

Any organization desiring to register a domain name must contact the administrator of the containing domain. For companies under .COM, or institutions under .EDU, refer to the document *Domain Administrators Guide* (RFC 1032, Network Information Center, SRI International, November, 1987)

Those organizations that don't have a domain name, but are reachable through Symbolics Dialnet, are provided with an umbrella domain under which they can register their own domains. This domain is `Dialnet.Symbolics.COM`. For example, the host Twinkie at Yoyodyne Industries might have the domain name of:

```
Twinkie.Yoyodyne.Dialnet.Symbolics.COM
```

Registration as a Dialnet domain is accomplished by installing Dialnet, the store-and-forward mailer, and the Domain system. The installation instructions for these products are documented elsewhere:

See the section "Installing Dialnet".
See the section "Installing and Configuring the Mailer".

After these are installed, send electronic mail to:

```
Customer-Reports@Riverside.SCRC.Symbolics.COM
```

To register, you should supply the following:

- The name of your company
- The name of the domain under `Dialnet.Symbolics.COM`
- The real name and the mail name of the individual responsible for the maintenance of the site
- A voice telephone number to reach this individual
- The name and Dialnet address of the server running Dialnet

You will receive return mail informing you that you have been registered.

Registration on both Internet and Dialnet is superfluous. If your machines run IP/TCP, that protocol will always be used in preference to Dialnet. Internet has a higher priority than Dialnet because IP/TCP is a more efficient means of communication.

The Launch File

The actual format of this file is more flexible than what is described here, but for ease of explanation, it has been simplified for this discussion. For full documentation on the file format, see *Domain Names - Implementation and Specification* (RFC 1035, Network Information Center, SRI International, November 1987).

In the Genera Domain Resolver, we use a "launch" file which is formatted like the standard domain files. This file is SYS:SITE;LAUNCH-DOMAIN-SERVER.TEXT.

A record takes one line and has three fields separated by one or more whitespace characters. Comments start with semicolons. The format of a record is:

```
<type> <domain> <auxiliary-information>
```

- The first field in a launch file record is the *type* field.
- The second field is the *domain* field.
- There might be a third field, *auxiliary-information*, whose meaning depends upon the given *type* field.

There are four record types. The record type is determined by the *type field* of a record. The four record types are:

Domain: The *domain* field contains the name of the domain which this resolver serves.

Primary: The *domain* field is the name of a domain for which this resolver is a primary resolver. The third field contains the name of a domain file for this domain.

Secondary: The *domain* field is the name of a domain for which this resolver is a secondary resolver. The third field contains the Internet address of the primary resolver for that domain. Upon startup, the primary resolver is asked for all information in this domain.

Dialnet: The *domain* field contains `Dialnet.Symbolics.COM`. The third field contains the name of a dialnet registry. Typical names are SYS:SITE;PRIVATE-DIALNET-REGISTRY.LISP and SYS:SITE;PUBLIC-DIALNET-REGISTRY.LISP. This is a Symbolics extension to the Domain system that allows us to use a file format which is already in use at many Symbolics sites.

The Internet Domain File

The actual format of this file is more flexible than what is described here, but for ease of explanation, it has been simplified for this discussion. For full documentation on the file format, see *Domain Names - Implementation and Specification* (RFC 1035, Network Information Center, SRI International, November 1987).

The domain file is referenced by a Primary record in the launch file. This is a standard file which is read with its domain name taken from the domain field of the Primary record of the launch file. The Domain system specifies the format of this file. A typical name for a domain file is SYS:SITE;WHATSAMATTA-DOMAIN.TEXT.

The first field in a domain file is the domain field. In many cases the domain name in this field is the name of a host. This field is optional. If it is omitted, it defaults to the value of the domain field in the previous record.

The second field is the class field. This is the class of protocol which is relevant for the given record. A class of IN means Internet, a class of CH means Chaosnet, and a class of DIAL means Dialnet.

The third field is the type field. The record types are described below.

Depending on the class and type of the record, there may be an RDATA field. The RDATA field is documented below.

Here are some notation conventions which can be used in this file:

. A freestanding dot refers to the current domain name.

.. Two freestanding dots represent the null domain name of the root.

@ A freestanding @ denotes the current origin. The origin is the name that is appended to names without trailing dots, and the value of "@".

\X Where X is any character other than a digit (0-9), it is used to quote that character so that its special meaning does not apply. For example, you can use "\." to place a dot character in a label.

( ) Use parentheses to group data that crosses a line boundary. In effect, line terminations are not recognized within parentheses.

" " A leading whitespace means the "last name read".

The Domain system allows a label to contain any 8-bit character. Although the Domain system has no restrictions, other protocols, such as SMTP, do have name restrictions. Because of other protocol restrictions, we recommend that you use only the following characters in a host name (besides the dot separator):

"A-Z", "a-z", "0-9", dash and underscore

All times are in units of seconds, and all integers are in decimal.

Domains referred to in this file are relative to the domain named in the main file. For an absolute domain name, append a dot to the end of the domain name. This means that if the launch file contains the following:

Primary Symbolics.COM sys:site;Symbolics-Domain.Text

Then the following translations occur in SYS:SITE;SYMBOLICS-DOMAIN.TEXT

File Entry	*Translation*
	Symbolics.COM
Riverside.SCRC	Riverside.SCRC.Symbolics.COM
Vermithrax.SCH	Vermithrax.SCH.Symbolics.COM
Think.COM.	Think.COM

The current origin is Symbolics.COM. Each name following will have Symbolics.COM appended to it. Thus, Riverside.SCRC becomes Riverside.SCRC.Symbolics.COM. When Think.Com. is read, the trailing period states that this is the full domain name.

The record types which may appear in the domain file and are discussed here are:

Type	Meaning
SOA	Start Of Authority
NS	Name Server
MX	Mail eXchanger
A	Address

SOA Start Of Authority. This record is traditionally the first record of the file. All other fields in this record are required. The *domain* field should contain the name of the domain.

Domain names in the zone files can be either of two types, absolute or relative. An absolute name is the fully qualified domain name and is terminated with a period. A relative name does not terminate with a period. The current default domain is appended to a relative name. The default domain is usually the name of the domain that was specified in the boot file that loads each zone.

The form of an SOA record is:

<name> <class> SOA <origin> <person> (

<serial>
<refresh>
<retry>
<expire>
<minimum>)

The Start Of Authority record designates the start of a zone. The zone ends at the next SOA record.

<name>	The name of the zone. @ is typically used here.
<class>	The protocol group.
<origin>	The name of the host on which the master zone file resides. Typically this is a relative name, as the zone will be owned by some host contained within it.
<person>	The mailbox for the person responsible for the administration of this domain. This is formatted like a mailing address except the @ that normally separates the user from the host name is replaced by a dot. Typically this is a relative name.

RDATA field. The fourth field of all resource records. The RDATA field of an SOA record contains some of the information relevant to administration of the domain, in order of appearance. Here is the information contained in RDATA:

<serial>	The serial number of the file is an unsigned 16-bit version number of the data in the file. This value should be manually incremented any time changes are made to the file.
<refresh>	The refresh interval is an unsigned 32-bit number. It is the time interval, in seconds, before a secondary name server should check with the primary server to see if the data in the domain should be brought up to date.
<retry>	The retry interval is an unsigned 32-bit number. It denotes the time interval, in seconds, after a failure that a secondary name server should wait before it attempts to refresh its information about the domain.
<expire>	The expiration interval is a 32-bit number. It is the upper limit of time, in seconds, that a secondary name server should allow before the data in the domain can no longer be considered authoritative. At that point a new update should occur.
<minimum>	The minimum interval is an unsigned 16-bit number. It is the minimum number of seconds that should be exported with any record from this zone (other than the SOA itself).

There should only be one SOA record per zone.

NS Name Server. This record form contains the name of a host which is acting as an authoritative name server for the domain. The form is:

<domain> <class> NS <server>

<name>	The name of the zone. @ is typically used here.
<class>	The protocol group.
<server>	The name of a host which is acting as an authoritative name server for the domain.

MX Mail Exchanger. MX records specify where to deliver mail for a domain. The <preference> field in an **MX** record contains a 16-bit integer preference value. The <host> field is the name of a host able to accept mail addressed to *domain* over *class* networks. The form of an MX record is:

<name> <class> MX <preference> <host>

<name>	The name of the zone. @ is typically used here.
<class>	The protocol group.
<preference>	A 16-bit integer preference value.
<host>	The host field host field is the name of a host that can accept mail addressed to domain over class networks.

There may be multiple MX records for a particular domain name. A preference value specifying the order a mailer should try multiple MX records when delivering mail may appear immediately before the host name. When mail is being sent to a domain, it may be sent to any host which has an MX record, but lower-numbered MX records will be tried first. If a path already exist to that host the Symbolics mailer will use that path first.

An Address. An address record contains the address of a host for a given class.

<host> <class> A <address>

<host>	The name of a host.
<class>	The protocol group.
<address>	The address of the host for the given class.

Diagnosing Problems with the IP/TCP Installation

For sites connected to the ARPA Internet, note that when the Internet network is enabled for the first time, during booting, the system has not yet read in the namespace updates that define the new gateway.

To solve this problem, boot the world and immediately give the Save World command. The world is saved with the updated information, and should work correctly when you next boot it.

For information on verifying whether a remote host is responding, see the section "Verifying Whether a Remote Host is Responding".

Reference Information on IP/TCP

This section contains background information for IP/TCP, and information for installing the Internet Domain Names System. It does not contain any installation instructions. For step-by-step instructions on installing the IP/TCP system, see the section "IP/TCP Installation".

For additional reference information on IP/TCP, see the document *Networks*, and read these topics:

"Internet Networks"

"Desirability of Network Protocols"

"TCP and UDP Protocols Supported by Symbolics Computers as Users"

"TCP and UDP Protocols Supported by Symbolics Computers as Servers"

"Format of Internet Addresses"

"Choosing a Network Addressing Scheme"

"References to IP/TCP Protocol Specifications"

Using IP/TCP on the Symbolics Computer

IP/TCP protocols are used automatically by the Symbolics computer whenever they are appropriate for performing some network service. For example, the TELNET protocol is used over the TCP medium when you use the Terminal program; the TCP-FTP protocol is used when remote files are opened; and the ASCII-NAME protocol is used by the Show Users command.

For more information on how to use Symbolics networking capabilities:

See the section "Using the Network".

See the section "Recovering From a Network Problem".

Using IP/TCP on the MacIvory

Note that one restriction exists when using IP/TCP on MacIvory. The Macintosh enables only one network using ARP (Address Resolution Protocol) to run at any given time. Due to the fact that IP/TCP is a network using ARP, you can enable IP/TCP either on the Macintosh or the Ivory, not on both. Additionally, you cannot enable a network on the Macintosh using ARP (such as Chaos) while using IP/TCP on the Ivory (note that the opposite is also true).

Commands are available for switching the control of the Ethernet cable from the Macintosh to the Ivory. These two commands are Ivory Menu Items that you can invoke from Lisp or the FEP:

Enable Network	Gives the Ivory control over the network.
Disable Network	Gives the Macintosh control over the network.

IP/TCP Protocols

The IP/TCP software package includes support for most of the standard IP/TCP protocols.

In addition, several protocols designed and developed by Symbolics run on TCP. These include: NFILE, 3600-LOGIN, and CONFIGURATION. NFILE provides FILE service over TCP; 3600-LOGIN provides LOGIN service over TCP; and LISPM-FINGER provides LISPM-FINGER service over UDP. For further information on NFILE, see the section "NFILE File Protocol".

For a complete list of IP/TCP protocols supported, see the document *Networks* and read these topics:

"TCP and UDP Protocols Supported by Symbolics Computers as Users"

"TCP and UDP Protocols Supported by Symbolics Computers as Servers"

IP/TCP Support for Subnetting

Symbolics IP/TCP supports subnetting. This section describes subnetting, and explains how to set up a site to use subnetting.

Subnetting Overview

An Internet address is divided into two parts: the network number and the host number. A Class A network uses the high-order byte for the network, and the low-order three bytes for the host. A Class B network uses the high-order two bytes for the network, and the low-order two bytes for the host. A Class C network uses the high-order three bytes for the network, and the low-order byte for the host.

Some Internet sites prefer to use subnetting, a practice in which some portion of the bytes traditionally used to indicate the host number is interpreted as a subnet number. For example:

Example of Class B Internet Address: 139.3.12.41

```
+--------+--------+--------+--------+
|10001011|00000011|00001100|00101001|
```

```
        +--------+--------+--------+--------+

Case 1: |<----network---->|<-----host------>|
Case 2: |<----network---->|<subnet>|<-host->|
Case 3: |<----network---->|<---subnet-->host|
```

In Case 1, no subnetting is done. Cases 2 and 3 are examples of subnetting. In Case 2, one byte is used as the subnet, and one for the host. In Case 3, 12 bits are used as the subnet and 4 bits are used as the host number.

Setting up a Site to Use Subnetting

In order for subnetting to work properly, it must be clear which bits are to be interpreted as the subnet and which as the host. This information must be entered into the namespace database, in the **user-property** attribute of the Internet network object.

The **user-property** attribute should be followed by INTERNET-SUBNET-MASKS. Next is a list whose elements are lists containing two strings: the address of a network that uses subnetting, and a representation of which bits should be masked, when trying to determine the host number. For example:

```
Case 2:

  INTERNET-SUBNET-MASKS (("139.3.0.0" "255.255.255.0"))

Case 3:

  INTERNET-SUBNET-MASKS (("139.3.0.0" "255.255.255.240"))
```

The address of the network is represented as four numbers separated by periods. The numbers are in decimal.

The mask that follows the address of the network is more tricky. It is also four numbers separated by periods, and represented in decimal. In the mask, each bit used in the subnet number should be a 1, and each bit used in the host number should be a 0.

```
        +--------+--------+--------+--------+
        |11111111|11111111|11111111|00000000|
        +--------+--------+--------+--------+
Case 2: |<----network---->|<subnet>|<-host->|
Mask:      255   .   255   .  255   .   0


        +--------+--------+--------+--------+
        |11111111|11111111|11111111|11110000|
        +--------+--------+--------+--------+
Case 3: |<----network---->|<---subnet-->host|
Mask:       255   .   255   .  255   .  240
```

If the network object is used to represent more than one network number, the **user-property** attribute can contain the mask for each network. For example:

```
INTERNET-SUBNET-MASKS (("139.3.0.0" "255.255.255.0")
                       ("115.22.0.0" "255.255.255.0"))
```

Note: Although the example above shows the network numbers on separate lines, when you type this into the Namespace Editor, the network numbers must all be on one line.

Routing

A host sending an IP packet has to determine whether to send the packet directly to a destination on the same network or to send it to another network. The sending host makes this determination by comparing the source address to the destination address. If the source and destination are on the same network, the host sends the packet directly to the destination. Otherwise, the host sends the packet to a gateway, which sends the packet to the destination.

Routing Within a Site

The Internet address class of a host determines an internal routing strategy for the subnet on which the host resides. If the subnetting scheme used at a site does not match the default subnet arrangement for that address class, you have to add a subnet mask to the Internet network object in the namespace. Additionally, you have to add the host objects representing each gateway to the namespace.

Specifying Subnet Masks

You have to specify a subnet mask if the field of the address used for selecting a subnet is differrent from the network field specified by the address class. When a host compares the source and destination addresses between witch it sends a packet, the subnet mask determines which bits are significant for the comparison. The subnet mask consists of two parts:

1. The official Internet subnet number of the network in question.

2. A mask with a 1 in every bit selecting a network or subnet. Note that if a site uses more than one range of address, you have to provide a subnet mask for each range.

You have to store subnet masks in the namespace under the user property `INTERNET-SUBNET-MASKS` of the Internet object.

Consider the following example: Symbolics allocates all addresses in the range 128.81.x.y, a class B address. Several subnets cover the entire address range, and leased lines in addition to Lisp machine gateways interconnect the different sites. You can establish addresses for this network by:

- Using the first two octets for selecting the corporate network.
- Using the third octet for selecting a subnet.
- Using the last octet for selecting a host on a particular subnet

In this example, the Symbolics corporate network requires the following mask: `"128.81.0.0" "255.255.255.0"`.

Determining the Subnet Mask for a Network

The subnet mask for a network consists of two parts:

- The first part of the mask consists of the Internet subnet number which you can derive from an address on the network.
- The second part of the mask consists of an Internet address containing a 1 in every bit selecting a network or subnet, and a 0 in every bit determining a host on a particular subnet.

For example, you can determine the first part of the mask (the Internet subnet number) for the class B address `128.81.38.232` by placing zeros in the last two octets of the address. The corresponding network number for `128.81.38.232` is `128.81.0.0`.

You can determine the second part of the mask for the class B address `128.81.38.232` by filling the first two octets with ones (since the first two octets select a network or subnet), filling the third octet with ones (since the third octet selects a subnet within the local network), and by filling the fourth octet with zeros (since the fourth octet selects a host on a subnet). The second part of the mask is `11111111.11111111.11111111.00000000` in binary, or `255.255.255.0` in dotted decimal notation.

Internet Domain Names

Introduction to Internet Domain Names

The Internet Domain Names system is a collection of specifications and procedures which implement the DOMAIN protocol, which is commonly used on the ARPA Internet. The DOMAIN protocol deals extensively with naming. It was created to address several problems.

One major problem that the Domain system addresses is the management of the ever-growing number of hosts on the Internet. When there were only a few hundred hosts, it was reasonable to keep a master file of hosts in a central location to be copied across the network periodically. As more and more hosts were registered, the Internet administrators found that they wanted to separate the hosts into smaller administrative units. Information about these hosts would then be main-

tained locally. As a result, the Domain system places these hosts in a tree-structured administrative system.

The second major problem that the Domain system attempts to address is the difficulty encountered when sending mail between different networks. Each network has a different naming scheme. These different naming schemes have hindered the interconnection of various networks. The Domain system attempts to allow connections between networks as diverse as Internet, CSnet, BITnet, UUCP, Symbolics Dialnet, and others.

For instance, in the past, mail addresses looked like:

- *user%host*.CSnet@CSnet-Relay.ARPA
- *random-host*!*uninteresting-host*!*host*!*user*@UCBVAX.ARPA
- adi/*user%host*.BITnet@WISCVM.ARPA

When addresses are automatically generated by various mailers, the results can be combined to make long and complex addresses.

If all the hosts involved are using the Domain system, all these mail addresses may be viewed as:

- *user@domain*

Symbolics implements the Domain specification described in several Requests for Comments (RFCs) available from the Network Information Center, SRI International. Symbolics implements the Domain specification on both TCP and Chaosnet. See the section "References to IP/TCP Protocol Specifications".

How the Domain System is Structured

Domains are administrative entities. There are no geographical, topological, or technological constraints on a domain. The hosts in a domain need not have common hardware or software, nor even common protocols. Most of the requirements and limitations on domains are designed to ensure responsible administration.

The Domain system is a tree-structured global namespace that has a few top-level domains. The top-level domains are themselves subdivided into domains. These domains can be further subdivided into yet more domains, and so on.

The administration of a domain requires controlling the assignment of names within that domain and providing access to the names, addresses, and list of valid services to users both inside and outside the domain.

The top-level domains are:

- **GOV** Government
- **EDU** Education

- **COM** Commercial
- **MIL** Military
- **ORG** Organization (an "other" category)
- **NET** Network administrative entities

Temporarily, the top-level domains also include:

- **ARPA** The current ARPA-Internet hosts

Additionally, the English two-letter codes identifying a country according to the International Standards Organization (ISO) Standard for *Codes for the Representation of Names of Countries* (ISO 3166, International Standards Organization, May 1981) can be used as top-level domains.

Sufficiently large companies can qualify for their own top-level domain. As of this writing, no company has attempted to qualify for a top-level domain.

How Domain Names Are Structured

The structure of a domain name is formally prescribed. Domains names are printed with each level of the domain name separated by a period. The Domain system knows nothing about hosts or sites; it deals only with names. The order of appearance in a domain name goes from the most specific to the most encompassing. For example, one of the Symbolics domain names is:

```
SCRC.Symbolics.COM
```

Based on the structure of the name, we can surmise that `SCRC` is the particular domain within Symbolics, and indeed it corresponds to a site in the Symbolics namespace. Continuing, we deduce that Symbolics is the name of the company that has a domain name of `Symbolics.COM`, and that `COM` is the top-level domain for commercial organizations. It is equally possible that `SCRC` is the name of a host. There is no way to tell *from the name* what `SCRC` is.

A host named "Rocky" in the Aerospace department at Whatsamatta University might have the domain name of:

```
Rocky.Aero.Whatsamatta.EDU
```

Looking from the most general to the most specific, this host is a part of the `EDU` domain. More specifically, it is a part of the domain `Whatsamatta.EDU`. Within the domain `Whatsamatta.EDU`, there is another domain—`Aero.Whatsamatta.EDU`. At this point, there is nothing to tell us if `Rocky` is a domain or a host in a domain. We know that `Rocky` is a host, because it is stated above. But you should always be aware of this potential ambiguity when reading domain names.

A domain name is just a name. The naming convention requires that some authoritative entity agree that it will be responsible for providing information about some domain and will guarantee that the information provided will follow the domain

conventions. There is nothing implicitly better, worse, different, or otherwise unusual about the number of segments in a domain name.

As a consequence of the above convention, periods are effectively reserved characters. The domain `Whatsamatta.EDU` should not be referred to as `Whats.a.matta.EDU`. The latter is in an entirely different domain. A name must contain either a character, a numeral, a dash, an underscore, or some combination of these elements. Domain implementations are currently required to be case-insensitive.

How Genera Uses Internet Domain Names

This section describes how Genera implements Internet Domain Names, and how they are related to the Namespace system. For related information, see the section "The Domain System and the Namespace System".

How do Symbolics computers find network-related information?

In the Symbolics networking environment hosts must be able to obtain certain types of information about hosts and users of the network. The Namespace system stores that information in its database, and provides it to hosts that request information. A typical site has one designated namespace server.

Along with the namespace database, Symbolics computers support the Internet Domain Names style of requesting and obtaining network-related information.

Symbolics computers look for naming information as follows:

- They first seek it in the namespace database.
- If the information is not found, and the request involves an Internet Domain Name, they seek the information from hosts on the network called Name servers.

What kind of sites benefit from Internet Domain Names?

This facility is useful for:

- Sites with one or more hosts that use IP/TCP and are connected to the ARPA Internet, or any Internet that uses Domain Names.
- Sites that use Dialnet.
- Any site that uses the Internet Domain Names style of addressing.

When is the Internet Domain Names facility used?

The Internet Domain Names facility is integrated with the generic network system's procedure for finding a path to a host. When a network service is requested from a remote host, the generic network system must find a path to that host. For example, when you send an electronic mail message, the "To" field can contain an Internet Domain Names style of name, such as:

```
To: Customer-Reports@STONY-BROOK.SCRC.SYMBOLICS.COM
```

The generic network system must find the network address of the host named STONY-BROOK.SCRC.SYMBOLICS.COM in order to send the message.

How does Genera find a host's network address?

The part of Genera that does this is called the *name resolver*. Specifically, it is code that is part of **net:parse-host**, which is used often by the generic network system. The name resolver first consults the namespace database for this kind of information. If the information is not found, and the name is an Internet Domain Names style of name (with periods separating the components), the name resolver uses the Internet Domain Names facility. These steps are described below.

How does the name resolver search the namespace for an Internet Domain style name?

In this case, the name resolver looks for a namespace whose **internet-domain-name** attribute is SCRC.SYMBOLICS.COM, and then looks for a host named STONY-BROOK in that namespace. If no such namespace is found, or no host is found in such as namespace, the resolver begins to seek the information from Name servers. A name must contain at least one period to be a candidate for this kind of resolution.

How does the name resolver seek the Internet Domain Names information?

The name resolver determines whether it has the requested information stored in a local cache; this would happen if it had already processed a similar request. This step saves the resolver from making an unnecessary search for information. If the information is not found in the local cache, the resolver seeks the information from another host on the network. The resolver makes a request of the central name resolver, if any host at the site provides the **:domain** service. If not, it makes a request of one of several designated hosts on the network known as Name servers.

How does the name resolver know if the site has a central name resolver?

When a host is booted, or the Reset Network command is given, the host looks in the namespace database in the current site for any hosts that provide **:domain** service. If so, the resolver always makes requests of the central name resolver instead of making requests directly of the Name servers. If no host provides **:domain** service, the host makes direct requests of Name servers. The host consults the **root-domain-server-address** attribute of the site object to find out the addresses of the servers for the top-level ("root") domain.

What namespace objects does the name resolver create?

Because so much of the network software depends on objects being present in the namespace, the name resolver was implemented to create a host object for hosts that were not already stored in the namespace, but were located via some Name server. For the host named STONY-BROOK.SCRC.SYMBOLICS.COM, a namespace called DOMAIN is created, if not already present. A host object named STONY-BROOK.SCRC.SYMBOLICS.COM is created in the DOMAIN namespace, if it is not already present.

Symbolics Computers as Central Name Resolvers

Name servers are hosts that provide a service to all hosts on the Internet. A *central name resolver* is a host that provides a service to all hosts at a site; that service is described here.

Some sites gain advantages when they designate a single host to perform most of the name resolution for the entire site. Each host at the site contains the name resolver software, but in this configuration that code does not make requests to Name servers on the network. Instead, it makes a request of the central name resolver host. Note that you can configure your site to have multiple hosts designated as central name resolvers.

A central name resolver receives requests from hosts at the site, and processes them by requesting the desired information from Name servers. When information is returned, the central name resolver shares it with the user host, and also stores it in a local cache. Thus, if a second host at the site requests the same information, the central name resolver can return it quickly, without resorting to another network request.

To designate a host as a central name resolver, you should add the following service attribute to its host object:

```
Service: Set: DOMAIN CHAOS DOMAIN
```

If the resolver supports IP/TCP protocols, you should also add the following:

```
Service: Set: DOMAIN TCP DOMAIN
```

Symbolics Computers as Name Servers

The name resolver lets a Symbolics computer go out to the network to request information from Name servers. In addition, Symbolics computers can be Name servers themselves.

When a Symbolics computer is designated as a Name server, it has a responsibility to provide information to other hosts on the network regarding hosts, users, and other network objects within its domain. When it is booted, it loads a file that defines its domain and some other configuration data. Much of the information that the Name server needs resides in the namespace database. The Genera implementation takes advantage of that, and does not require that the Name server duplicate information already stored in the namespace. When the Name server needs information not present in the namespace, it can be stored elsewhere. The file SYS:SITE;LAUNCH-DOMAIN-SERVER.TEXT contains the pathnames of any additional data files.

A computer that is designated to be a Name server *for the ARPA Internet* must support IP/TCP, because it must be capable of communicating with other hosts on the Internet using IP/TCP.

Note that a Symbolics computer can be a Name server even if it is not connected to the ARPA Internet and does not support IP/TCP. For example, a site that supports only Chaosnet protocols could still use Internet Domain Names to name users and hosts. All that is required is that each host on the network is capable of requesting Name resolution and that the designated Name server is capable of storing and providing the information necessary to resolve Internet Domain Names.

It is not necessary that a Symbolics computer acting as a Name server have the **:domain** service attribute in its host object.

Internet Domain Name Namespace Attribute

During installation you specify the Internet Domain Name to be associated with the namespace in which local hosts are registered, by editing the **Internet Domain Name** attribute of the namespace object that represents the local namespace itself. All hosts that are named within that namespace then inherit the Internet Domain Name that is entered in the namespace object.

For example, the SCRC namespace object might have this attribute:

```
Internet Domain Name:  SCRC.Symbolics.COM
```

SCRC|JUNCO is a host named Junco in the SCRC namespace. Junco inherits the Internet Domain Name of its namespace, so its Internet Domain Name is:

```
Junco.SCRC.Symbolics.COM
```

In some cases a host in that namespace is not in the same Internet domain. An individual host can override the Internet domain of its namespace by entering a value in the **Internet Domain Name** attribute of its host object. In this example the host SCRC|GRACKLE has the Internet Domain Name Grackle.MIT.EDU.

```
Internet Domain Name:  Grackle.MIT.EDU
```

The **Internet Domain Name** attribute of the host object is used solely to override the attribute of the namespace object.

The Domain System and the Namespace System

In many ways, the Domain system and the Namespace system attempt to solve the same problems. Both the Namespace system and Domain System attempt to deal with the issue of naming. Both systems deal with a collection of names that refer to a grouping of machines. In the case of the Namespace system, this collection is called the namespace. In the case of the Domain system, we shall refer to this as a domain or a subtree. However, it is important not to draw too close an analogy between the two.

It might appear that both systems map to "administrative entities". Actually, the Domain system returns attributes that are connected to names. The Namespace system goes beyond the Domain system in describing the hosts, users, printers, and networks within an entity known as a Site. There is nothing in the Domain

system that is equivalent to a Site. Humans make the connection between a name and an administrative entity like a site; the Domain system software deals *only* with names.

The major difference between the Symbolics Namespace system and the Domain system is that the space containing the set of all Namespace names is flat, whereas the Domain system is organized as a hierarchy. From one perspective, this hierarchy can be viewed as a tree-structured administrative hierarchy.

Any site which has Symbolics computers must use the Namespace system. Communication with other Symbolics machines within a site can occur without use of the Domain system if the Symbolics machines are in the Namespace system. Any site which wants to communicate with other sites must use the Domain system. If there are non-Symbolics machines at your site and you want to communicate with them via IP/TCP, you should run the Domain system. If you are running Dialnet and Genera, you must use the Domain system.

The Domain system and the Namespace system appear to have information which overlaps. In point of fact, you cannot describe information via the Domain system that is also represented in the Namespace system. In other words, the Namespace System is *always* asked first, and it *always* wins any argument about the validity of any piece of data. If a query about a host that is mentioned in both the Namespace system and in the Domain system occurs, the information from the Namespace system will be used.

There is only one way of assuring that the information in the Namespace system and the Domain system don't conflict: by making a Symbolics computer the primary domain resolver for machines that are in the Symbolics namespace at your site. If this is done, the namespace information will be used to complete the domain information. If this is not done, data integrity will be compromised, since you must manually update all host information in both the Namespace system and the Domain system at the same time.

If you have machines that are not part of the Symbolics namespace, you should have a Symbolics machine serve as the piece of the domain tree that corresponds to the Symbolics namespace, and let any other machine deal with other parts of the domain tree. There is no useful way a machine can be a server for only part of a namespace/subtree. Note that nowhere in this discussion have we mentioned "site", only namespace.

You cannot have a partial representation of the hosts in the Namespace system and the remainder in a domain server elsewhere. Partial representation of information in one domain server and the rest in another domain server is also not allowed. Confusion occurs when there is *not* a single authority for a block of names, when one server has one piece of the namespace/subtree and some other server has the rest of the namespace/subtree. *This restriction is not a characteristic of the namespace implementation nor of any domain implementation. Rather, it is a fact common to naming schemes.* If partial information were allowed, it is easy to see that problems would arise as soon as one server's information differed with another's. There *must* be an authoritative server in any naming scheme.

If your organization already has a domain resolver running on another system, you have two options:

- Move the domain resolver to a Symbolics machine.
- Create a new sub-domain containing the Symbolics machines with a Symbolics machine as a domain resolver.

Summary of the Internet Domain Names Facility

Name resolver
: This code is used to resolve network names, such as turning a host name into the correct network address for that host. The code is part of **net:parse-host**. If a name (such as a host name) contains periods, it is an Internet Domain Names style of name. In these cases, the name resolver checks to see if the namespace database contains the information. If not, the name resolver makes a network query for the needed information. The name resolver queries a central name resolver if any are designated at the site. If not, it queries one of the Name servers directly. The host can look at the **Root-Domain-Server-Address** attribute of the site object to find out the addresses of the top-level Name servers. If that attribute of the site object is empty, and no central name resolvers have been designated for the site, then the name resolver cannot resolve the requested name.

Central name resolver
: This is a host that the site depends upon to perform name resolution for all Symbolics computers at the site. A central name resolver is designated by having one or two service attributes for **:domain** service in its host object.

```
Service: Set: DOMAIN CHAOS DOMAIN
Service: Set: DOMAIN TCP DOMAIN
```

To resolve a name, a host first checks the namespace database. If the name is not present in the namespace, the host submits a request to the central name resolver, using the **:domain** protocol. The central name resolver checks its local cache to see if it contains the requested information. If not, it makes a request to a designated Name server. The central name resolver decides which Name server to ask by looking at the **Root-Domain-Server-Address** attribute of the site object. If that attribute of the site object is empty, the central name resolver cannot perform the resolution.

Name server
: This is any host that provides information on names and addresses to other hosts, using the DOMAIN protocol. Symbolics computers are capa-

ble of being Name servers. The server software is a separately loadable system. Note that a central name resolver serves the hosts within the site, but a Name server also serves hosts outside of the site. Sites can configure one Symbolics computer to be both a central name resolver and a Name server.

IP/TCP Security Considerations

You can control security through the **secure-subnets** attribute in the site namespace object. IP/TCP software automatically denies access to Internet hosts unless you have listed the host (or the network on which the host resides) requesting the connection in the **secure-subnets** attribute of the site object. If you have existing sites that you have not created entries for in the **secure-subnets** attribute of the site object, you must edit the site object and add this attribute. Otherwise, the server denies access. (Note that this does not affect Chaosnet; Chaosnet trusts all hosts when no Chaos entry exists in the **secure-subnets** attribute.)

Configuring the Internet Security Mechanism for Your Site

Security is controlled by the **secure-subnets** attribute of the site namespace object. The **secure-subnets** attribute can contain both an Internet entry and a Chaos entry. In order for a secure IP/TCP server to accept a connection, the address of that host or the network of that host must be listed in the Internet entry of the **secure-subnets** attribute of the site namespace object.

Sites that prefer to allow access to all Internet hosts should enter the value "ALL" in the Internet entry of the **secure-subnets** field.

For new sites, or existing sites whose site object never had an Internet entry for the **secure-subnets** attribute: when you install IP/TCP, no Internet hosts will be trusted by that machine, and secure IP/TCP servers will refuse connections from Internet hosts. The solution is to edit the site namespace object and add a **secure-subnets** attribute whose first element is INTERNET. The second element should be "ALL" to trust all hosts (this restores your site to its previous behavior with respect to Internet security), or one or more Internet subnet addresses to explicitly state which subnets/networks should be trusted.

Here are some examples. The following entry allows access to all Internet hosts on subnets within 128.81.0.0:

```
Secure Subnets: INTERNET 128.81.0.0
```

The following entry allows access to all Internet hosts on 128.81.0.0 and to host 192.10.41.25:

```
Secure Subnets: INTERNET 128.81.0.0 192.10.41.25
```

The following entry allows access to all Internet hosts:

```
Secure Subnets: INTERNET ALL
```

IP/TCP Debugging Tools

In addition to the suggestions contained in this section for debugging IP/TCP problems, you might want to alter some of the default parameters. See the section "Tuning IP/TCP".

Verifying Whether a Remote Host is Responding

You can verify whether a remote host is responding by using the **tcp:send-icmp-echo** function.

tcp:send-icmp-echo *host* &key *:length :timeout* *Function*

Sends a message to the remote host and returns **t** if the remote host is responding, or **nil** if it is not. Note that if you are attempting to contact a remote host for the first time, this function returns **nil** because a cached address resolution for the remote host does not yet exist. Subsequent attempts using **tcp:send-icmp-echo** return **t** if the remote host is responding.

Using Peek to Check IP/TCP Status

Once IP/TCP is installed and loaded, you can check its status with the Peek facility. For a description of Peek, see the section "Using Peek".

Peek is available from the System menu, or by using `SELECT P`. Click the mouse on the Network heading.

By clicking on Meters, under INTERNET, you see the values of various meters collected by the Symbolics computer. The display is continually updated.

The following variable is useful for debugging IP/TCP:

tcp:*ip-debug-flag* *Variable*

Controls whether various unexpected conditions within IP cause notifications. Its normal value is **nil** (no notifications).

Adding Other Internet Protocols

Application protocols for TCP and UDP are accessed through the generic network system. The user side of the protocol should be defined with **net:define-protocol**, the server side with **net:define-server**. See the section "Defining a New Network Service".

If you are defining a private protocol to be used within your site only, we advise you to use a port number between 256 and 1024. If your protocol is intended to be used outside your site, we recommend that you apply to the ARPA Network Information Center for a valid port number. For the address, see the section "References to IP/TCP Protocol Specifications".

TCP

TCP supports the generic **:byte-stream** and **:byte-stream-with-mark** mediums. Use the function **tcp:add-tcp-port-for-protocol** to associate a TCP port number with the protocol name you have defined.

tcp:add-tcp-port-for-protocol *protocol-name tcp-port-number* *Function*

Associates a TCP port number with a protocol. *protocol-name* is a keyword symbol, *tcp-port-number* a number.

```
(tcp:add-tcp-port-for-protocol :smtp 25.)
```

UDP

UDP supports both the generic **:datagram** medium and the specific **:udp** medium. Use the **tcp:add-udp-port-for-protocol** function to associate a UDP port number with the protocol name you have defined.

The **:udp** medium supports one medium-specific **net:define-server** keyword:

:connection The value of this keyword is a symbol to be bound to the UDP connection "stream".

tcp:add-udp-port-for-protocol *protocol-name udp-port-number* *Function*

Associates a UDP port number with a protocol. *protocol-name* is a keyword symbol, *udp-port-number* a number.

```
(tcp:add-udp-port-for-protocol :tftp 69.)
```

IP/TCP Implementation Notes

The Symbolics computer implementation of IP/TCP conforms to the published standard. For more information on the documents describing the standard specification of IP (Internet Protocol) and TCP (Transmission Control Protocol), see the section "References to IP/TCP Protocol Specifications".

Tuning IP/TCP

IP/TCP is shipped in a state appropriate for general use on a local area network. Certain parameters can be altered to tune the performance of the IP/TCP system for specific situations. If you change any of these, you should carefully monitor your network's performance for unexpected side effects.

tcp:*adaptive-tcp-retransmission-enabled* *Variable*

Controls whether TCP retransmission uses a fixed retransmission interval or attempts to adapt the retransmission interval to the response time of the remote host. **t** is the default value (enable adaptive retransmission). Adaptive retransmission algorithms are stable only if the variance of the response time is not too large. If the variance is small enough, setting this to **t** significantly increases the performance of the TCP connection.

tcp:*background-interval* *Variable*

Controls the interval between executions of the IP background routing function. Its normal value is 3600 (1 minute).

See the variable **tcp:*dead-gateway-ping-interval***.

See the variable **tcp:*live-gateway-ping-interval***.

tcp:*dead-gateway-ping-interval* *Variable*

Controls the rate at which gateways believed to be down are probed to see if they have come up. Its normal value is 3600 (1 minute). Probing happens only when the IP background routing function runs.

See the variable **tcp:*background-interval***.

tcp:*default-window-size* *Variable*

Controls the TCP window offered to remote hosts. Its normal value is 20000 octets.

tcp:*ip-default-max-packet-size* *Variable*

Controls the maximum length of IP datagrams that can be sent through gateways. Its normal value is 576 octets.

tcp:*live-gateway-ping-interval* *Variable*

Controls the rate at which gateways believed to be up are probed to see if they have gone down. Its normal value is 36000 (10 minutes). Probing happens only when the IP background routing function runs.

See the variable **tcp:*background-interval***.

tcp:*max-window-size* *Variable*

Controls the maximum TCP window to be used on a remote host. Its normal value is 20000 octets.

tcp:print-recent-tcp-headers &optional *count* *Function*

count is the number of TCP headers to display. If it is not given, all those stored are displayed. The headers are displayed in reverse chronological order. **tcp:*record-tcp-debugging-info*** must be set to **t** to enable recording of headers.

tcp:*record-tcp-debugging-info* *Variable*

Controls whether TCP header information for the last 64 segments should be recorded for the use of the **tcp:print-recent-tcp-headers** function. This can be used to debug network problems. Its normal value is **t** (recording enabled).

tcp:*tcp-connect-timeout* *Variable*

Controls the timeout when making a TCP connection. The system will continue to try for this long, and then stop trying.

tcp:*tcp-idle-probe-interval* *Variable*

Controls the rate at which TCP "idle probe" and "zero window probe" messages are sent. The normal value is 7200 (2 minutes). "Idle probe" messages are sent over connections over which there has been no traffic during the interval. They contain only the appropriate ACK. They are sent in the hope of causing an RST or ICMP Destination Unreachable if the connection has actually died.

tcp:*tcp-response-timeout* *Variable*

Controls the time after which a TCP connection is abandoned if the remote host does not respond. Its normal value is 3600 (1 minute).

tcp:*tcp-retransmit-interval* *Variable*

Controls the initial retransmission interval for TCP connections. Its normal value is 120 (2 seconds). If adaptive retransmission is enabled, the retransmission interval is adjusted to match the remote host's actual response time.

See the variable **tcp:*adaptive-tcp-retransmission-enabled***.