XEROX

BUSINESS SYSTEMS

Systems Development Department

To:	Gateway Administrators	Date:	March 23, 1981 12:13 AM
From:	Hal Murray	Org:	SDD/Comm
Subject:	Gateway Parameters	Filed:	[IVY]<Portola>Mesa6>GateParameter.press

This memo describes the details of the parameter file used by the Mesa 6 version of the Alto Gateway program and the Rubicon version of the Pilot (Pup) Gateway.

1. Conventions for Parameter files

The Gateway program uses the Tools routines to parse parameter files. Each section of a parameter file begins with a line that looks like:

[*SectionName*]

Each section is composed of a collection of lines with the following format:

tag: *value*

A blank line marks the end of a section. *Beware, a line without a* : *is also considered to be an end marker.* In most places, case is ignored.

There are no general provisions for comments within a section. You can put almost anything you want between sections. Most routines in the Gateway that look in parameter files will treat a line whose *tag* begins with a ; as a comment. Normally, when a routine in the Gateway recognizes a line from the parameter file as something significant, it prints it back out in a slightly different format so that you can easily tell how it interpreted the input data. This is helpful when you are trying to make sure that you didn't leave out a : which would cause it to terminate scanning prematurely. Most routines that scan a parameter file print a message when they encounter an unrecognized tag. This can be used as a comment that appears in the typescript file.

2. User.cm

There are normally two sections used in User.cm. The first tells the Gateway program the name of it's parameter file. It looks like:

```
[Indirect]
ParameterFile: hostname.txt
```

Where *hostname* is the name of your Gateway.

If you are running in Alto mode, you need a second section. It looks like:

```
[TOOLS]
Bitmap: [x: 32, y: 128, w: 544, h: 350]
ToolsFont: SysFont.strike
```

You may, of course, adjust the size and/or location of the bitmap. These parameters work.

If you are running under Pilot/Tajo, you can start the Gateway automatically when the volume is booted by including the following lines:

```
[Tajo]
Initialize: PilotGateway
```

If you want to boot the volume without starting the Gateway, use the N switch. If you don't have this line (or you have booted with the N switch), and now want to start the Gateway, Run

PilotGateway/I. If you forget the I, all of the various Tools that are included in the Gateway will come up Active.

3. Gateway section (Alto mode)

The `[Gateway]` section of the parameter file contains the details of the hardware configuration and various parameters that pertain to the operation of the whole machine rather than a particular server.

If you have a debugger installed on your disk, the Mesa system normally reserves some memory for the debugger to use for it's bitmap The following line returns the space to the Gateway program.

```
Kill Debugger Bitmap: TRUE
```

Starting things from the Exec with `Gateway/k` will do the same thing. If the value is anything except `TRUE`, the line doesn't do anything. If you have a 256K machine it is reasonable to give some memory to the Debugger. It sure makes running the Debugger go faster. If you don't have a debugger on your disk, you don't need this line.

The system buffer pool is shared by all of the processes using buffers. If you want to change it, for example to save memory, add a line like the following:

```
Buffers: number-of-buffers
```

The default number of buffers for the system buffer pool is 25. If you have fewer than 25 buffers in the pool, you will probably run into performance problems if your Gateway has several phone lines or more than one Ethernet. The system also allocates an additional 4 buffers for each ChainedEthernet, 2 for each normal (unchained) Ethernet, and 2 for each SLA line.

In order to keep the typescript file from filling up the disk, add a line like the following:

```
Typescript: number-of-pages
```

This line starts another process that watches the size of the typescript file. When the size of the file overflows the specified number of pages, the output pointer is reset to near the front of the file. The pages are not deleted -- you can still see their old contents by using FTP to move the file to another machine. Unfortunately, you can't see anything beyond the current write pointer in the display window. 100 pages fits in Bravo or Laurel.

The system heap is shared by all of the processes in the machine. It is used to hold the name server cache, the Boot Server table, the parameters for HostWatcher, and all the information needed to keep the Tools windows displayed correctly. A line like the following allocates a large block of storage for the heap.

```
ExpandHeap: number-of-pages
```

If you don't expand it during initialization, you may run into troubles with memory fragmentation after the Gateway has been running for a while.

The Gateway program normally needs special microcode to work properly. To load it, include a line like the following:

```
Microcode: filename
```

Don't forget to put the file on the disk. `MesaGateCPChain1822.br` contains support for 3 Chained Ethernet boards, and the BiSync portion of the CommProc microcode. It also supports an 1822 interface to the Packet Radio. `MesaGateCPChain.br` includes all of the CommProc microcode, but it only supports 2 Chained Ethernets. Use `MesaGateEIAChain.br` if you have an EIA board. It supports 3 Chained Ethernets, and the BiSync portion of the EIA microcode.

In order to tell the Gateway the numbers of the networks that it is connected to, you must include one of the following lines. If you have more than one Ethernet, you must include one line for each network.

```
ChainedEthernet: net host board
Ethernet: net host board
```

Most Gateways will use the `ChainedEthernet`. *`net`* is the network number that the board is connected to, *`host`* is the host number of the machine, and *`board`* is the board number. The normal ethernet board is number 0. The host number must be the same on all Ethernets. If you specify a `ChainedEthernet`, you must load the appropiate microcode. If you are just building a boot server, don't load any microcode and don't specify any IO devices. Note that `ChainedEthernet` and `Ethernet` both refer to the 3 mb Ethernet. There is no controller available to connect a 10 mb Ethernet to an Alto.

To use a phone line, you need one of the following two lines:

```
        CommProc: net host lines
        EIA: net host lines
```

`net` is the network number of the SLA net. It will normally be 7. *`host`* is the SLA host number of the machine. If you are building a new Gateway, NetSupport.WBST will assign you a new SLA host number. Don't coufuse the SLA host number with the Ethernet host number. *`lines`* is the number of lines that the Gateway program should use. If you specify 0 lines, the Gateway program will setup the LCBs and a Cleanup Proc so that you can safely get to the debugger, but it will not actually poke any of the hardware.

The Alto Gateway actually contains the code to forward OISCP format packets. In order to turn it on, you must include the following line in your parameter file.

```
        Activate OISCP Router: TRUE
```

You should probably contact SDD before you use this feature.

Since Gateways usually run unattended, the following line can be used to turn off the display when the Gateway program starts.

```
        Turn Off Display: TRUE
```

If you put it at the end of the Gateway section, you can watch the other messages as they are printed. (You can, of course, turn the display back on with Ctl-LeftShift-D.)

For reference, here is the appropiate section of Twinkle's parameter file. Only the Packet Radio parameters have been removed.

```
        [Gateway]
        Kill Debugger Bitmap: ~TRUE
        Buffers: 25
        MICROCODE: MesaGateCPChain1822.br
        TYPESCRIPT: 100
        ExpandHeap: 25
        ChainedEthernet: 3 364 0 ; First Ethernet - second floor of 35
        ChainedEthernet: 5 364 1 ; Second Ethernet - bldg 34
        ChainedEthernet: 6 364 2 ; Third Ethernet - first+third floor of 35
        CommProc: 7 15 5 ; SLA via CommProc
        Turn Off Display: TRUE
```

Contact Larry Stewart or Hal Murray (or look at the code) if you need information on the Packet Radio parameters. See section 5 of this memo for information on passwords for dial up lines.

4. Gateway section (Pilot mode)

The `[Gateway]` section of the parameter file contains the details of the hardware configuration and various parameters that pertain to the operation of the whole machine rather than a particular server.

The system buffer pool is shared by all of the processes using buffers. If you want to change it, for example to save memory, add a line like the following:

```
        Buffers: number-of-buffers
```

The default number of buffers for the system buffer pool is 25. If you have fewer than 25 buffers in the pool, you will probably run into performance problems if your Gateway has several phone lines or more than one Ethernet. The system also allocates an additional 4 buffers for each Ethernet.

In order to keep the typescript file from filling up the disk, add a line like the following:

```
Typescript: number-of-pages
```

This line starts another process that watches the size of the typescript file. When the size of the file overflows the specified number of pages, the output pointer is reset to near the front of the file. 100 pages fits in Bravo or Laurel. The process that watches for the file to get too big also forces the bits out to the disk occasionally, so you should include a `Typescript` line even if you want the file to get huge.

The system heap is shared by all of the processes in the machine. A line like the following allocates a large block of storage for the heap.

```
ExpandHeap: number-of-pages
```

This isn't very important in Pilot.

In order to tell the Gateway the numbers of the networks that it is connected to, you must include one of the following lines. If you have more than one Ethernet, you must include one line for each network.

```
Ethernet: board net
EthernetOne: board net
```

`Ethernet` is for 10mb Ethernets. `EthernetOne` is for the older 3mb (Alto) Ethernets. *`board`* is the board number*. `net`* is the network number that the board is connected to. The first board is number 1. For an EthernetOne, the host number comes from the switches on the boards.

The Pilot Gateway normally turns on the OISCP forwarder. In order to bypass this, you must include a line like the following in your parameter file.

```
Activate OISCP Router: FALSE
```

Since Gateways frequently run unattended, the following line can be used to turn off the display when the Gateway program starts.

```
Turn Off Display: TRUE
```

If you put it at the end of the Gateway section, you can watch the other messages as they are printed. (You can, of course, turn the display back on with Ctl-LeftShift-D.) On a D0, it isn't very important to turn the display off.

For reference, here is the appropiate section of Glypnod's parameter file.

```
[Gateway]
Ethernet: 1 74
EthernetOne: 1 60
EthernetOne: 2 25
Buffers: 50
ExpandHeap: 30
Typescript: 1000
```

5. Passwords for Dialup lines

A password mechanism is available to prevent unauthorized people from using the Xerox internet by calling into a port that is set up to automatically answer. There are three steps needed to enable this feature.

First, use the PasswordTool (it's normally inactive) to compute the encrypted form of the desired password. The encrypted form includes the time, so don't be surprised when you don't get the same answer if you try a second time.

Second, update the `[Gateway]` section of the parameter file on the Dialin Gateway to mark the appropriate line(s) as needing a password, and enter the encrypted form of the password. The following lines will set up line 3 to use the password "testing":

```
CommProc: ....
DialinPassword: 172427 11553 125426 113132 10327 14616 142642 150070
```

```
        LineNeedsPassword: 3
```
The `LineNeedsPassword` entry must come after the `CommProc` or `EIA` line.

Third, add the line "`SetupDialoutPassword: TRUE`" to the `[Gateway]` section of the parameter file on the Dial out Gateway. Whenever the Gateway is restarted, this will create a Dialout Password Tool so that you can enter the password. You must (re)enter the password each time the Gateway is restarted since it is not stored in any file.

6. TimeServer section

In order to activate the TimeServer, you must specify 3 parameters. Twinkle's TimeServer section looks like this:
```
        [TimeServer]
        ZONE: +8:00
        DST: 121,305
        CORRECTION: -2
```
The `ZONE` parameter is used to specify the local time zone. + is west of Greenwich. The Pacific zone is 8, and the Eastern zone is 5. The number after the : is for the few places that use half hour alignment. It is the number of minutes to be added to the hours. The `DST` parameter specifies the first and last days of daylight savings time. The above numbers are correct for most places in the U.S.. If you don't want to use daylight savings time, set them both to 367. The `CORRECTION` parameter is used to fine tune the clock. Its units are in seconds per day, and a positive number makes the local clock go faster. It isn't supported on Pilot yet (it is ignored), but you still need to specify it to activate the Pup Time Server. If you have a machine with a flakey clock (for example, a D0 with an old Misc board), an easy way to keep it from polluting the net is to leave out the `CORRECTION` parameter. An easy way to do that is to change it to be "`*CORRECTION`".

If you are just setting up a boot server, please do not activate the time server unless you are going to take the time to check the clock. The problem is that it's clock will drift, then the bad information will get used by a nearby gateway as it restarts after a crash, and then it will propagate around the net when the next gateway crashes.

7. TimeChecker section

In order to help calibrate clocks, the Gateway program includes a process that compares the local time with that from another time server. It puts a line into the typescript file every hour. It needs a section that looks like:
```
        [TimeChecker]
        Target: target
```
Twinkle and Maxc are good machines to use since their clocks are well calibrated. If you are many hops from Palo Alto, you should probably use a closer machine.

If your machine and the target machine have both been up for several days, you should be able to compute the most significant portion of the clock correction. It takes a week or two to get the low order bit. Synchronizing with WWV or the phone company time server also helps. The typescript file wraps around reasonably often, so you should write down the starting information if you are trying to make a long run.

8. BootServer section

The Gateway program also includes a boot server. This is where the programs come from when you boot an Alto while holding down the backspace key. The boot server also provides microcode to get a D0 going if its disk is not ready or the microcode on it is broken. The MesaNetExec can also be used to load Othello from a boot server. It does this by retrieving microcode and a germ from a boot server, and then starting the germ such that it will load Othello over the Ethernet. Similar facilities are also provided for Dorados. The `[BootServer]` section looks like:

```
        [BootServer]
        comment: number boot-file-name
        comment: number boot-file-name
        comment: number boot-file-name
        comment: number boot-file-name
        .....
```

The *`comment`* is currently ignored. Someday that field may be used to distinguish various properties of boot files, for example, does it have a software checksum.

The master list of boot file number-name pairings is kept on [Maxc2]<BootFiles>BootDirectory.txt.

Boot servers exchange information with their neighbors every hour or so. If they notice a newer version of a boot file (that is in their list) they will try to retrieve a copy of it. This is how new versions of DMT.boot propagate around the net. Unfortunately, this only works if all intermediate Gateways have a copy of all of the interesting boot files. To circumvent this restriction, the boot server makes a long range scan every 24 hours or so. A long range scan checks for new files on boot servers on every net within 3 hops.

Boot files with numbers of 100000 or over do not propagate. It is a good idea to keep a file named *`HostName`*`.boot` with a number of 100000 on your machine so that you can test the boot server.

When the boot server is retrieving a new copy of a boot file (or you are storing it with FTP), the file is first written into a temporary file. This is to keep from trashing things in case something fails before the transfer finishes. This means that you must maintain suficient free space on your disk for a second copy of the largest boot file that will get updated. The largest Alto file is 256 pages, but Othello.pb is over 400 pages. You also need another 100 pages for the system.

When you store a new copy of the parameter file (via FTP), the boot server will see it arriving, and recycle itself in case you have updated the `[BootServer]` section. You do not have to restart the Gateway for it to notice the new parameter file.

9. FTP Server

There is an FTP Server built into the Gateway. Anybody can retrieve any file, but you need to get past the password checker before you can store (or delete, or rename) a file.

First, the FTP Server searches a table of name-password pairs. To add an entry to the table, include a line like the following in the `[FTPServer]` section of your parameter file.

```
        Password: username number..number
```

You can add as many entries as you need. You can use the PasswordTool (its normally inactive) to find the necessary octal numbers.

If the FTP Server doesn't find a match in the table, it tries to use Grapevine to verify passwords. As well as providing a valid name-pasword pair, the username must also be a member of the group (distribution list) *`hostname`*`.internet`. Grapevine expects all names to include a registry such as PA, or WBST. If you want to specify a default, include a line like the following in the `[FTPServer]` section of your parameter file.

```
        Default Registry: registry
```

`registry` will be appended to the username provided by FTP if the username doesn't already contain a period. IFS and Maxc will happily ignore a registry if you provide one, so it's also easy to set up your disks so that your name already includes the registry. My disks all think my name is Murray.PA.

10. HostWatcher section

There is also a HostWatcher packaged inside the Gateway. It tries to make contact with a server every 15 minutes. If the up/down state of the server has changed, it will send a message to a specified individual, list of individuals, or distribution list. The following message is typical:

```
Date: 12-Jan-81 20:05:56 PST
From: HostWatcher on SoEasy
Subject: RoyalArches is up
To: Murray.PA
cc: Murray.PA

RoyalArches is up: Alto Gateway of 11-Jan-81  0:00:25 up 0:12:56.
The last time it was up was 12-Jan-81 16:50:44 PST.
   255 (41%) up.
    26 (4%) rejecting.
   339 (54%) not responding.
```

There are two types of lines in the `[HostWatcher]` section of a parameter file. The first type doesn't cause any specific action. They just setup information that will be used later. The information lines have the following format:

```
Troubles: mail-system-name
To: list-of-recipients
cc: list-of-recipients
Full: list-of-recipients
```

They can occur in any order and as often as necessary. They are normally repeated and intermixed with action lines. Each individual (or distribution list) in the *`list-of-recipients`* must explicitly include the registry. You should be sure to include the `Troubles` line. If you have more then one, only the last one will be used. It is passed the the mail system as the Sender property, and that is who the mail system will send messages to if it can't deliver a message to somebody.

Each host that you want to watch needs one server line. The server lines have the following format:

```
Gateway: Gateway-name
Chat: telnet-server-name
FTP: FTP-server-name
MAIL: mail-server-name
Spruce: printer-name
Librarian: librarian-name
```

When the scanner encounters a server line, it adds an entry to a table that includes the data from the most recent information lines. If you want to watch both the FTP and Mail servers on an IFS, include 2 lines.

Here is an abbreviated copy of the `[HostWatcher]` section of Twinkle's parameter file:

```
[HostWatcher]
Troubles: Murray.PA
To: RWeaver.PA, Boggs.PA, Taft.PA
cc: Murray.PA
Full: Murray.PA
Chat: Maxc1
Chat: Maxc2
To: RWeaver.PA, Taft.PA
Chat: DLS
To: RWeaver.PA, Kolling.PA
FTP: Juniper
To: RWeaver.PA, Boggs.PA, Taft.PA
FTP: IVY
To: Mallory.PA, Murray.PA
Gateway: SoEasy
```

When HostWatcher notices a server go up or down, it sends a message to the *`list-of-`*

`recipients` in the `To` line, with copies to the `list-of-recipients` in the `cc` line. A server that does not respond is not considered down unless HostWatcher can still contact the last Gateway that was used to contact the server the previous time. If HostWatcher notices that a server is full (by matching the error text from a built in list) it sends a message to the `list-of-recipients` in the `Full` line, with copies to the `list-of-recipients` in the `cc` line. If no `Full` line was specified, it doesn't send any message.

Gateways are a special case for HostWatcher. Messages will be sent about Gateways if the network topology changes in an interesting way. This happens, for example, if a phone line used as an alternate path to the target Gateway goes up or down.

When you store a new copy of the parameter file (via FTP), HostWatcher will see it arriving, and recycle itself in case you have updated the `[HostWatcher]` section. You do not have to restart the Gateway for it to notice the new parameter file.

Beware: You can get a lot of mail from HostWatcher if the network is confused. Don't use if if you don't like junk mail. I have tried to sort out the stupid cases, but it still makes mistakes when a Gateway goes up or down just as HostWatcher is poking a server on the other side of the Gateway.

11. TrapBadPups section

The Gateway program contains a pile of kludgery to process Pups that have a bad software checksum. These are frequently caused by bad RAM chips in Alto Ethernet interface boards since the CRC doesn't detect errors they make.

When a Pup with a bad software checksum arrives, it is put on a special queue. If they arrive more frequently than once every 10 minutes on the average, they are discarded. Another process appends a message into the typescript file and optionally sends a message to a list of people. The following is a sample message:

```
Date: 11-Mar-81  1:43:07 PST
From: Murray.PA (on RoyalArches)
Subject: Bad Checksum info
To: Murray.PA
cc: Cude.PA

11-Mar-81  1:42:49  *****  Bad software checksum on Pup from net 74
Bad pup number 1 has a checksum of 74161, but it should be 73160
Encap:  177777 177777 177777      0      0  24046   1000
Header:    226      0      0  22760  36000      0      2  36332      0      2
Body:      460  51404   1060  51404   1460  51402   2460  51402   3060  51402
          3460  51401   4060  51402   4460  51402   5060  51404   5460  51404
          6060  51403   7460  51404  10460  51404  11060  51404  12060  51401
         12425 155000  13060  51404  13460  51403  14060  51406  15460  51404
         16060  51404  20060  51404  20460  51404  22460  51405  23460  51403
         26060  51402  26460  51403  27060  51403  30060 155000  30460  51403
         31060  51403  31460  51403
```

In order to interpret the information, you have to know what type of Ethernet the Pup came from. If it is a 3mb Ethernet, then only the last 2 words of the encapsulation are interesting. Net 74 is a 10mb Ethernet -- all 7 words are relevant. The above example appears to be a broadcast of a routing Pup from Glypnod (24046 74#332#). Be careful when trying to interpret the information in the packet since it is known to be damaged.

To activate the mailing option, add the following section to your parameter file:

```
      [TrapBadPups]
      Troubles: mail-system-name
      To: list-of-recipients
      cc: list-of-recipients
```

When a Pup with a bad pup arrives (and gets past the too-often filter), a message will be sent to the `list-of-recipients` in the `To` line, with copies to the `list-of-recipients` in the `cc` line.

Beware: You can get a lot of mail if a machine is broken. Don't use if if you don't like junk mail.

Revision history:

March 10, 1981, Split out from AltoGateway memo