# CONVEX
# SPP CXTS V1.6.1
# Release Notice

Document No. 081-025630-007

August, 1995

**CONVEX Press**
Richardson, Texas USA

# CONVEX
# SPP CXTS V1.6.1
# Release Notice

Document No. 081-025630-007



Printed in the United States of America

# Contents

# Release Notice

This document provides information about the SPP CONVEX Expert Troubleshooting System (CXTS) V1.6.1.

## SPP CXTS components

The SPP CXTS consists of the following components:

- CXTS server (`cxts_server`)
- Notification Browser (`notification_browser`)
- Automatic mailer (`cxts_send_mail`)

The `cxts_server` module is the foundation for CXTS. It provides the interface between the SPP system processes and the other CXTS modules. The notification browser window provides a graphical interface that allows the user to define events and trigger thresholds for these events. See the "Notification browser window" section on page 5. The automatic mailer module ensures that the appropriate personnel receive notification of event triggers by automtically sending e-mail to predefined addresses. See the "Using sendmail" section on page 12.

## Distribution

The distribution package for this release of CONVEX CXTS consists of this document, distribution media for the software, and the CONVEX CXTS User's Guide. The specific contents of the software and documentation distribution are described in the following table.

**Table 1** Contents of this Distribution

| Item | Quantity | Type | Part Number | Description | Format |
|---|---|---|---|---|---|
| 1 | 1 | DAT | 081-010015-007 | CONVEX SPP CXTS V1.6.1 | DAT |
| 2 | 1 | Doc | 081-025630-007 | CONVEX SPP CXTS V1.6.1 Release Notice | hardcopy |

## Reporting problems

If you encounter a problem using CONVEX CXTS, it is very important to gather system data at the time the problem is observed. The engineering organizations use information returned from the field to analyze and correct problems identified in the field. When a problem cannot be reproduced in the engineering lab, the collected data determines the corrective action.

Use VIPER or PRTS to report bugs associated with CXTS processes. When using VIPER, be sure to select `SPP CXTS` from the `Select Product` window under the `Product Info` option of the `Setup` menu.

# Using SPP CXTS V1.6.1

## Installing SPP CXTS

To install SPP CXTS V1.6.1, make sure that you are `su` or `root` and perform the following steps:

**Step 1** Make sure that the CXTS software package (including the `cxts_install` script) is loaded with the test station diagnostics. If not, following the procedure for loading the software from the Test Station Diagnostic installation procedure.

**Step 2** At the command prompt, enter the following:

```
cxts_intall
```

**Step 3** The install script asks the following question:

```
Are you single user at this time? [y|n]
```

Answer appropriately.

**Step 4** If `sendmail` has been previously set up, the installation is complete, and the script displays the following:

```
Setting up sendmail at this time
Setting up cron for sppuser at this time
Setting up /etc/inittab at this time
```

Do not perform the remaining steps.

**Step 5** If `sendmail` requires set up, the script will display the following:

```
Setting up sendmail at this time
What is your NIS domain name? (example: convex.com)
```

Answer the question by entering your domain name.

**Step 6** Answer the following question by entering either your name server or mailhost (e.g. convex):

```
What is the name of your name_server or mailhost?
```

**Step 7** The script then displays the following:

```
What is the ip address of your name_server or mailhost?
```

Enter the IP address (e.g. 130.168.66.1). If you do not know the IP address, see your system administrator.

**Step 8** The script then displays the following:

```
Setting up cron for sppuser at this time
Setting up /etc/inittab at this time
```

This completes the installation. The installation procedure sets the `/etc/inittab` file so that SPP CXTS starts automatically at boot up.

## Starting the notification browser

The `Notification Browser` allows you to define event triggers and event actions. CXTS loads default values at start up. However, it is highly recommended that the SSE use the `Notification Browser` during initial installation to make sure the event triggers and actions are set correctly.

To open the `Notification Browser` window type the following at the command prompt:

```
cd /spp/cxts/bin

notification_browser
```

Defining events and trigger thresholds is described on the "Notification browser window" section. If you change notification when `cxts_server` is running, you must send `SIGHUP` to `cxts_server` so that it will re-read the `notification.setup` file.

## Notification browser window

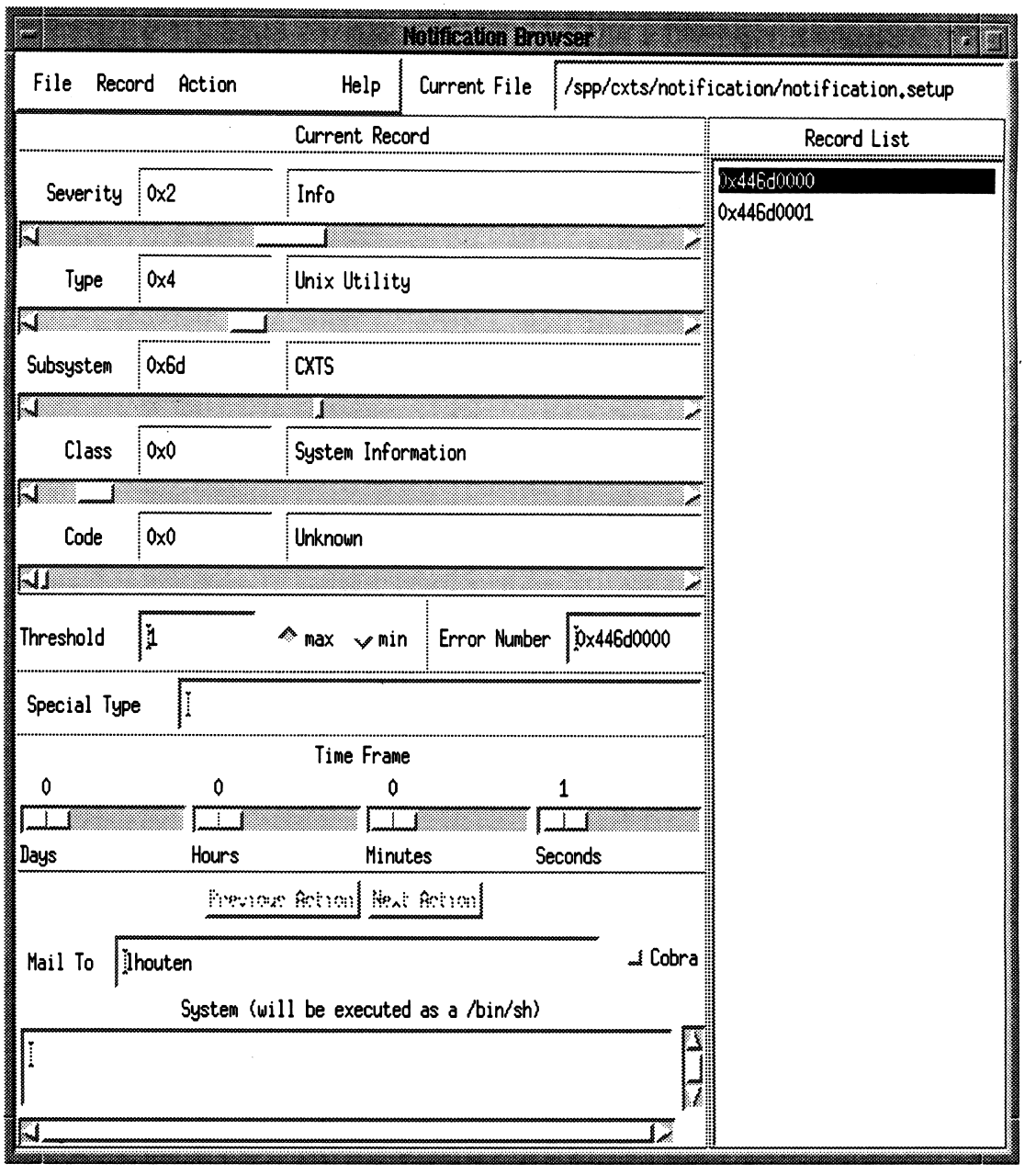
Automatic notification is invoked when a specified notification trigger threshold is reached. This happens when a certain event type occurs a specified number of times within a specified time window. An event is an occurrence or incident containing information that may be important to maintaining or troubleshooting the system.

The `notification.setup` file contains control parameters that define the notification-trigger threshold, the recipient of the notification, the message delivery route, and the commands that will be executed for each notification event. The file consists of *notification records*. Each record defines the combination of filters and thresholds that apply to an event. The records also define what is to be done as a result of the event. Each record is set using the `Notification Browser` window.

When triggered, automatic notification creates a message and sends it to the appropriate audience. Not all notification messages are of interest to the same groups. The TAC may be interested in some events, the system manager in other events, and CONVEX Engineering in still other events.

Multiple, separate recipients can be defined for a specific triggering event. The audience may differ according to the definition of the threshold. By updating the entries in the definition of a trigger threshold, you can specify where notification messages are sent.

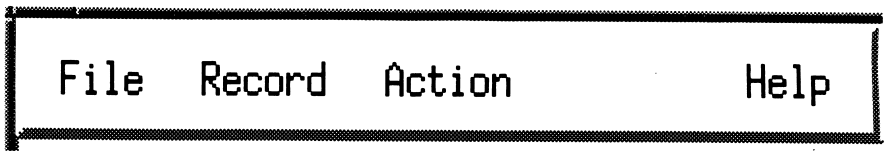
**Figure 1** Notification Browser window



## The Menu bar

The menu bar has four options: `File`, `Record`, `Action`, and `Help`. When you click on one of these options, a drop-down menu appears. Each drop-down menu, except `Help`, lists one or more actions. You invoke one of these actions by clicking on it.

**Figure 2** Menu bar




- `File`—Use this option to manipulate the setup files. The actions listed under the `File` option are:
  - `Show Comments`—opens a text window in which you can enter comments about a file.
  - `Write Current File`—updates the current file shown in the `Current File` field of the browser window with the changes you have made.
  - `Read File`—opens the `Read_File` window which allows you to load the browser window with parameters from a selected file.
  - `Write File`—opens the `Write_File` window which allows you to update a selected file.
- `Record`—Use this option to create, copy, or delete an event record. The actions listed under the `Record` option are:
  - `Create`—allows generating a new event record.
  - `Copy`—copies an event record.
  - `Delete`—deletes an event record.
- `Action`—Use this option to create, copy, or delete an action to be performed for a record. The actions listed under the `Action` option are:
  - Create—allows generating a new event action.
  - Copy—copies an event action.
  - Delete—deletes an event action.
- `Help`—This option opens this help file.

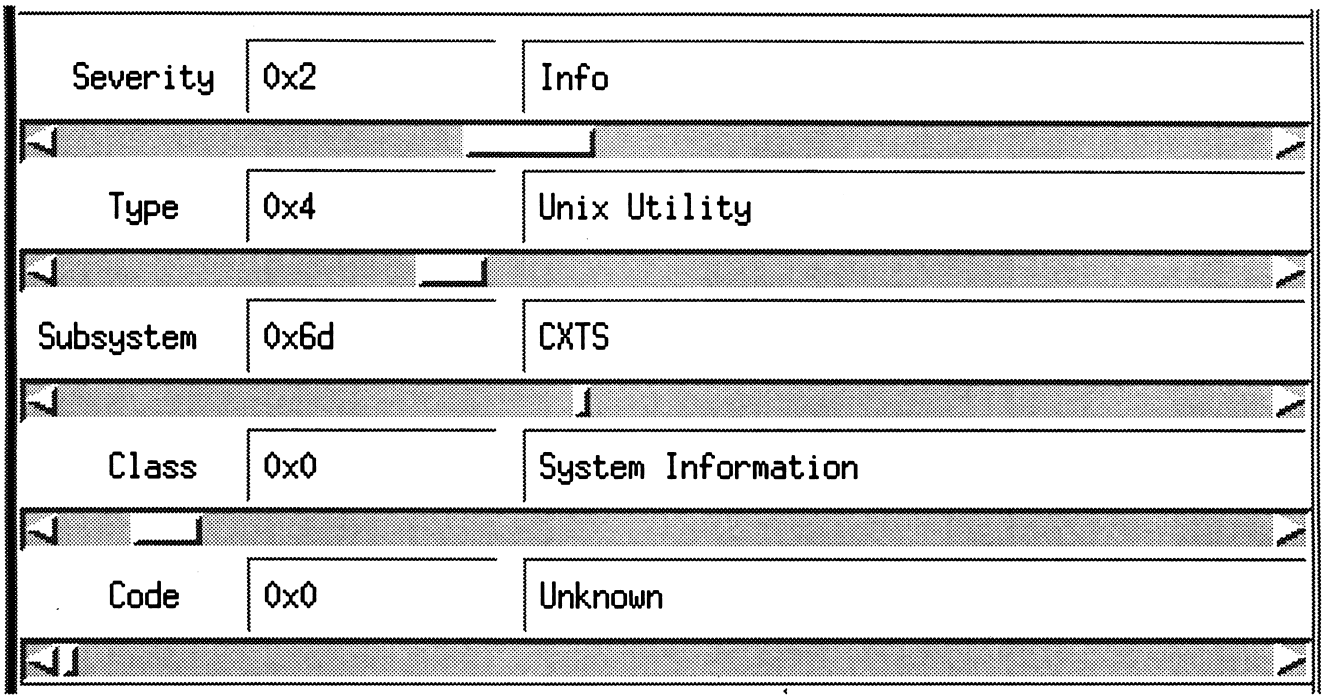
## Event records

When an event occurs on an SPP system, it is sent to the `event_logger` on the test station. `Event_logger` logs the event to a file and forwards it to `cxts_server` (see the "Notification browser window" introduction for the definition of an event). The `cxts_server` checks the event to see if notification is set up for its type.

An event is defined by adjusting the value in any of the five classification fields. These fields are:

- `Severity`
- `Type`
- `Subsystem`
- `Class`
- `Code`

**Figure 3** Event definition fields




Enter data into these fields by moving the slide bar just below the field display. Each of the slider bar buttons in the `Notification Browser` window operates in the same way. To move the slide bar, click on the slider with the left mouse button and hold the button while moving the slider to the left or right. When the slider is in the new position, the field code changes to the new value. For each of the five fields, the code and a description of the code is displayed above the slider bar. The arrows on either end of the bar allow you to increment through the field one value at a time.

The `Error Number` field is the event definition, and is the sum of the values in the classification fields. Whenever you change the value of one of the five fields, the `Error Number` reflects the change. The error number is also displayed in the `Record List` window.

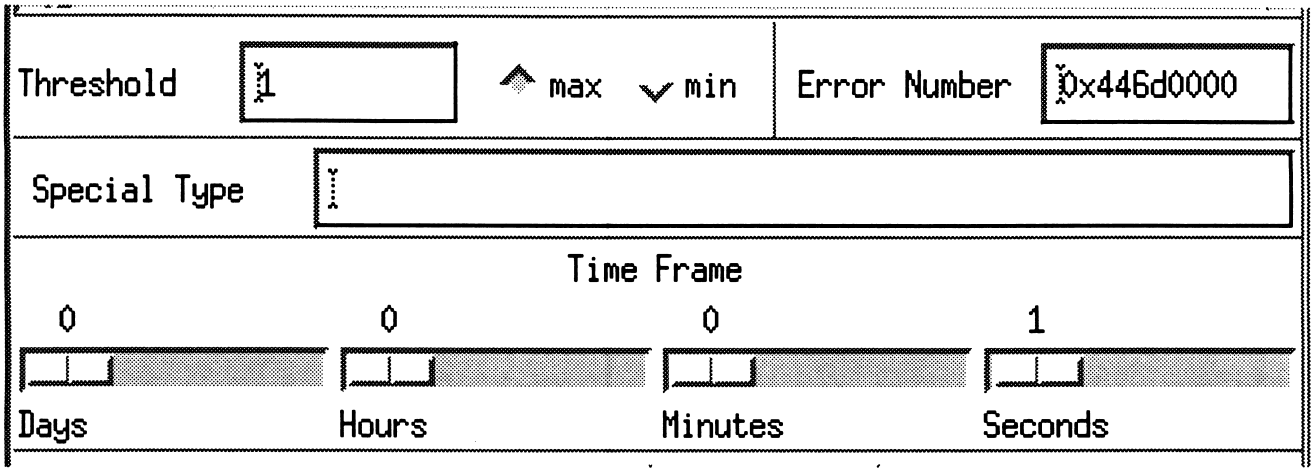
If any of the five fields is set to "Don't Care", or a value of -1, the field is ignored. In this way, you can select all events of a certain classification by selecting "Don't Care" for the other fields.

## Event threshold

The notification threshold may be set to trigger on a minimum count or a maximum count. To set the trigger:

1. Enter the desired number of occurrences in the `Threshold` field.
2. Click either `max` (maximum) or `min` (minimum).
3. Set the `Time Frame` by using slide bars, one each for days, hours, minutes, and seconds. Using the slide bars is described in the `Event Records` section.

**Figure 4** Threshold definition fields

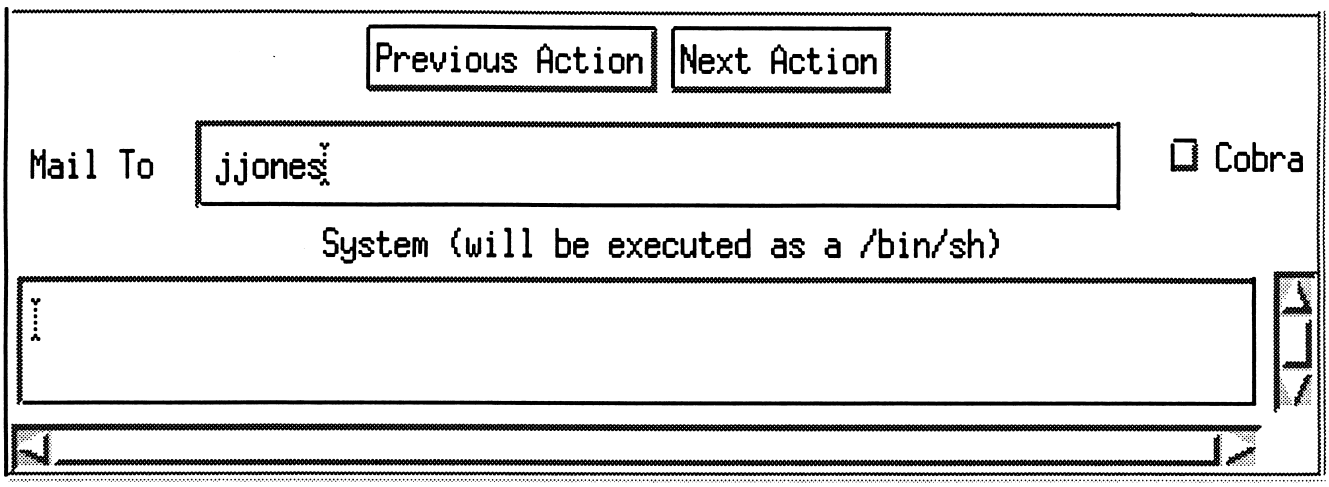


## Event Actions

Whenever the trigger threshold is reached, `cxts_server` issues system-level (shell) commands defined in the `System` box. Specific commands can be associated with the event. Different system commands can be configured to redirect their output to the notification message file. The file that is sent by notification is `/spp/cxts/notification/message`.

The `System` text box is used to include other information in the notification message. `stdout` and `stderr` can be appended to the end of the notification message. Multiple commands can be entered when separated by a semicolon or carriage return.

**Figure 5** Action definition fields




The following examples illustrate the use of commands:

To append the current CXTS journal file:

```
cat /spp/cxts/journal/journal.jnl >>
/spp/cxts/notification/message
```

To append the last 20 event log entries:

```
tail -20 /spp/data/event_log >>
/spp/cxts/notification/message
```

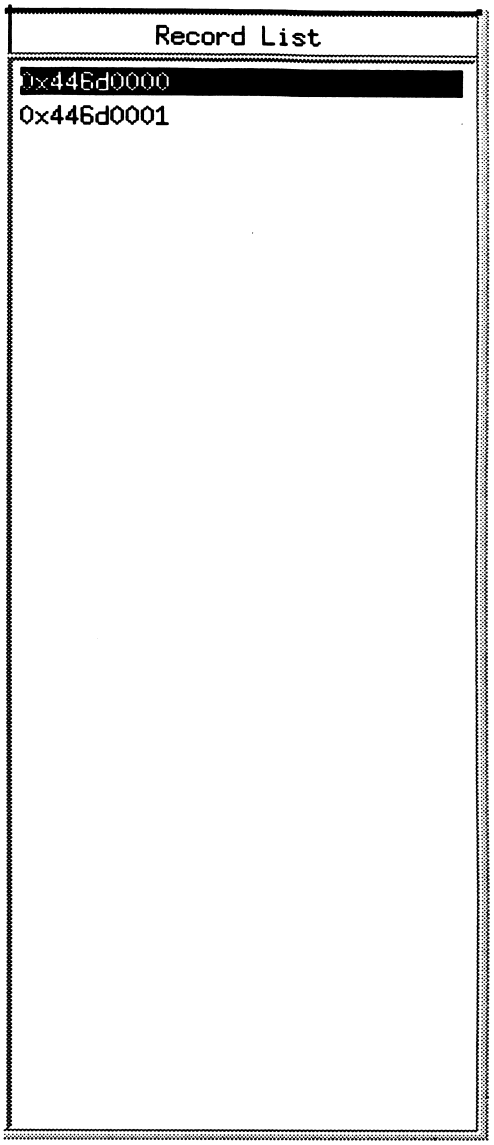
Multiple actions can be associated with each record. To create a new action, click on the `Action` option in the menu bar and select `Create`. When a new action is created, the `Previous Action` and `Next Action` buttons become active. These buttons allow you to "flip through" the actions to select one for an event record. An action may also be copied and then modified to generate a new action. To copy an action, click on the `Action` option in the menu bar and select `Copy` from the menu. You may also delete an action by selecting `Delete` from the `Action` menu.

The `Mail To` field lists the recipients of the event information. To enter mail recipients, click in the `Mail To` box and type the appropriate mail address. Multiple, separate recipients can be defined for a specific event by entering their mail addresses separated by a space.

## Record list

The `Record List` box lists all records in the `notification_browser.setup` file. Clicking on a record loads it into the `Notification Browser` window, where it can be edited.

**Figure 6** Record List window




## Data files

There are two data files used by the `Notification Browser`. The first one is the setup file mentioned in the "Notification browser window" introduction on page 5. This file is found in the directory `/spp/cxts/notification`. The other is the file `field.names`. This file is used to determine the classification field names. Each numeric value of the classification fields can be assigned a name to give the number more meaning. This file is found in the directory `/spp/cxts/bin`.

## Using sendmail

The `cxts_install` script sets up the `send_mail` routine. Normally, you will not have to manually set it. If however, you need to set up `sendmail`, perform the following procedure:

## Note

**The following procedure is not normally required.**

**Step 1** Change the permissions of `sendmail` by entering the following:

```
chmod 4755 /usr/lib/sendmail
```

**Step 2** Ensure that the `/usr/lib/aliases` file has the predefined and available aliases:

- `cxts_tac` `cxts_tac@convex.com`
- `cxts_eng` `cxts_eng@convex.com`

**Step 3** Ensure that the `/usr/lib/sendmail.cf` file has been correctly setup.

Set the **domain_name** in the following line of the `sendmail.cf` file to your local domain name:

```
# macro L defines the "local domain" to which you connect
directly for SMTP mail
DLdomain_name
```

Set the **mail_host** in the following lines of the `sendmail.cf` file to your `mail host name`:

```
# UUCP relay for unresolved! addresses (via SMTP)
DWmail_host
```

Set the **mail_host** in the following lines of the `sendmail.cf` file to your `mail host name`:

```
# SMTP relay for unresolved @ addresses
DSmail_host
```

**Step 4** Copy the `sendmail.cf` file to `/usr/lib`

**Step 5** Enter the following at the command prompt:

| | |
|---|---|
| **sendmail -bz** | (freeze the configuration file) |
| **sendmail -bi** | (get aliases ready) |
| **sendmail -bd -q15m** | (start in daemon mode every 15 minutes) |

**Step 6** Add **mail host** to the `/etc/host` file

**Step 7** Set up `crontab` by adding `sppuser` to the list in `/usr/lib/cron/cron.allow`.

Enter the following at the command prompt as `sppuser`:

```
crontab /spp/cxts/bin/VIPER_crontab
```

## Setting up the modem

The HP modem must be set correctly before the automatic pager can work. Perform the following procedure to set up the modem:

**Step 1** Make the nodes for the dialin and dialout lines. You must be root.

```
mknod /dev/cua00 c 1 0x204001
mknod /dev/cul00 c 1 0x204001
mknod /dev/ttyd00 c 1 0x204000
```

**Step 2** Add these lines at the end of the `/usr/lib/uucp/Devices` file.

```
#
Direct cul00 - 19200 direct
ACU cul00 cua00 19200 hayes96t ## use this if tone dialing
#ACU cul00 cua00 19200 hayes96p ## use this if pulse dialing
```

**Step 3** Add these lines at the end of the `/usr/lib/uucp/Dialers` file.

```
#
# Hayes Ultra96 Smartmodem -- This entry is set up for the
  configuration
# Convex Computer Corporation uses on it's SPP test station
# "hayes96t" is for tone dialing
# "hayes96p" is for pulse dialing
#

hayes96t =,-, "" \dATV1Q\d\r OK\r \dAT\r\c OK\r
\d\pATDT\T\r\c
   CONNECT
hayes96p =,-, "" \dATV1Q\d\r OK\r \dAT\r\c OK\r
\d\pATDT\T\r\c
   CONNECT
```

**Step 4** Add this line to `inittab`.

```
a0:3:respawn:/etc/getty -h ttyd00 19200
```

**Step 5** After installing the modem, enter the following:

```
cu -s 19200 -l/dev/cul00 -m dir
```

**Step 6** When you see the connected message, enter the following:

```
at&fm0e0v0x0q1s37=9s0=1&c1&d3&w0 <cr>
```

This will setup the modem and disconnect you.

## Using pager

The SPP `cxts_pager` is an automatic paging system that allows you to page any number of people. The pager allows you to use either a pulse or tone telephone system. You can set the number of times the page is sent and the number of people paged. The `cxts_pager` is described in Table 2

**Table 2** Arguments for `cxts_pager`

| | |
|---|---|
| [-p (pulse) \| -t (tone)] | Indicates the type of telephone system to which the modem is connected: either pulse or tone. |
| [/dev/_____] | Sets the path (use /dev/cul00). |
| [# pages] | Sets the number of times the page is sent. |
| [paging pause] | Sets the time delay during page in seconds. |
| [hang up pause] | Sets the time between pages |
| [pager number ...] | Sets telephone numbers to page |

**Example:**

The following example depicts a typical pager command:

```
cxts_pager -t /dev/cul00 1 30 5 9,,18005551212,,,,,,,,12345679,,,888
```

In this example the type of telephone system is tone (-t), the path is `/dev/cul00`, the number of times the page command is executed is one, the time delay during pages is 30 seconds, the time between pages is five, the remainder of the command consists of the telephone number (9 1 800 555-1212), the PIN number (1234567), and the message code (888). The commas add delay between the numbers for proper connection. Use the number of commas in the example as a starting point and adjust them for your particular area.

## SPP system monitoring

Two programs exist to provide system-uptime monitoring: `pulse_client` and `pulse_server` (referred to as the client and server, respectively). The client runs on the SPP and establishes a network connection to the server which runs on the test station. While the SPP is running, the client periodically sends messages to the server. The server expects to receive messages continuously. If the server does not receive a message within a specified period, or if the network connection is lost, the server logs a message into the `event_logger`.

Using the `Notification Browser` window, you can define an action for each event type.

### Setting up the server

Normally the `server` is set up on the test station when the "diag" tape is installed. When the test station boots, the server starts and runs continuously, requiring no attention. However, if you must start the server, make sure the following line is in `/etc/inittab`:

```
puls:2345:respawn:/spp/bin/pulse_server
```

### Setting up the client

You must set up the client, since it is not set up in advance. To set up the client on the SPP perform the following steps:

**Step 1** Make sure the network connection between the test station and the SPP is working properly.

**Step 2** Make the following entries in `/etc/inittab`:

`puls:2345:respawn:/etc/pulsed -h test_station` (substitute the name of system for `test_station`)

The server uses `test_station` to look-up the internet address in `/etc/hosts` when initiating the connection.

## About the client

The `pulse_client` has the following arguments:

**Table 3** Arguments for `cxts_client`

| | | |
|---|---|---|
| **-h** host_name | required | Provides the name of the host on which the server is running. The name is used by the server to look up the internet address in `/etc/hosts` when initiating the connection. |
| **-v** *version number* | optional | The version of the OS - If not provided, the program uses `uname()` to get the version |
| **-p** `port_number (0xa2c8793c)` | optional | Used to identify the port on which the server is located. If `port_number` is set, it must match the other port number in the server. |
| **-s** `seconds (60)` | optional | Specifies the period of time between messages sent to the server. |

## About the server

The `pulse_server` has the following option:

**Table 4** Argument for `cxts_server`

| | | |
|---|---|---|
| **-p** `port_number (0xa2c8793c)` | optional | Is set to match the one used in the client. If this parameter is set in this program, it must be set to match the other port number. |

When the server is running, it sends events to the `event_logger` on the test station, based on the information that it gets from the client.

The events are:

| | |
|---|---|
| `0x446d0000` | This is sent when the client makes its initial connection. The string that is passed with this event contains the version of the OS that has started. |
| `0x446d0003` | The CONTINUE event occurs when the client has previously been in communication with the server, has lost the connection, and is reconnecting. The string that is passed with this event contains the version of the OS. |
| `0x446d0002` | There are three separate actions that can cause a LOST message to be issued:<br>1. If an error occurs in the read call, the text explains the error (output from strerror (errno)).<br>2. If 0 bytes are read, it means the client died or closed its socket. The event text says "lost connection."<br>3. If a time out occurs, then twice the pulse period has gone by with no message. The event text says "Timed-out while waiting for OS pulse."<br>In all three cases, communication with the client has ceased, and the OS is presumed to be down. |