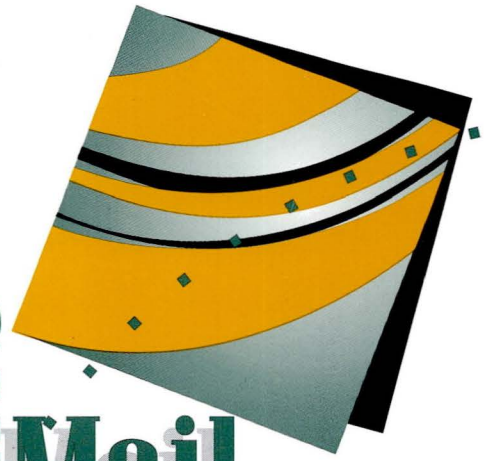


CONVEX

PRTSMail User's Guide

Fifth Edition

CONVEX
PRTSMail User's Guide

Order No. DSW-600

Fifth Edition
January 1994

CONVEX Press
Richardson, Texas
United States of America

CONVEX
PRTSMail User's Guide

Order No. DSW-600

Printed in the United States of America

Revision information for

CONVEX
PRTSMail User's Guide

Edition	Document No.	Description
Fifth	700-044030-004	Released January 1994 with version 2.2 of the PRTSMail server. Added information about `status bug` and CXSOFT customers.
Fourth	700-044030-003	Released May 1993 with version 2.1 of the PRTSMail server. Added information about the `date` modifier to the `status` command.
Third	700-044030-002	Released February 1993 with version 2.0 of the PRTSMail server.
Second	700-044030-001	Released November 1992 with version 1.0 of the PRTSMail server.
First	700-044030-000	Initial release, October 1992.

Contents

Figures

Tables

Preface

Purpose and audience

CONVEX customers have long demanded to be able to obtain the status of Problem Reports (PRs) they submitted with the `contact` utility.

The *PRTSMail User's Guide* describes how to access PRTSMail and how to interpret the information it returns. PRTSMail is an electronic mail interface to the CONVEX Technical Assistance Center's Problem Report Tracking System (PRTS) database. PRTSMail addresses the demand to access the PR status information via electronic mail with a simpler interface than the ACCES (Automated CONVEX Customer Enquiry System) database.

Notational conventions

This section discusses notational conventions used in this book.

`Bold monospace` In command examples, text shown in **`bold monospace`** identifies user input that must be typed exactly as shown.

`Monospace` In paragraph text, `monospace` identifies:

- Command names
- System calls
- Data structures and types

In command examples, `monospace` identifies command output, including error messages.

In command syntax diagrams, text shown in `monospace` must be typed exactly as shown.

Italic In paragraph text, *italic* identifies:

- New and important terms
- Titles of documents

In command syntax diagrams, *italic* identifies variables that must be supplied by the user.

{ } In command syntax diagrams, text surrounded by curly brackets indicate a choice. The choices available are shown inside the curly brackets and separated by the pipe (|) sign.

The following command example indicates that you can enter either `a` or `b`:

```
command {a | b}
```

[ ] In command syntax diagrams, square brackets indicate optional data.

The following command example indicates that definition of the variable *output_file* is optional:

`command` *input_file* [*output_file*]

...	In command syntax, horizontal ellipsis shows repetition of the preceding item(s). The following command example indicates you can optionally specify more than one *input_file* on the command line. `command` *input_file* [*input_file* ...]
KEYCAP	In paragraph text, text shown in KEYCAP indicates keyboard keys you must press to execute the command. For example, RETURN refers to the carriage return key. Two KEYCAP terms separated by a hyphen indicate two keys that you must press simultaneously. For example, CTRL-d indicates that you must press the d key while holding down the CTRL key.

Note

A Note highlights supplemental information.

Ordering documentation

To order the current edition of this or any other CONVEX document, send requests to:

CONVEX Computer Corporation
Customer Service
P.O. Box 833851
Richardson TX 75083-3851 USA

Include the order number or the exact title, as listed on the front cover.

Technical assistance

If you have questions that are not answered in this book, contact the CONVEX Technical Assistance Center (TAC).

- Within the continental United States, call 1 (800) 952–0379.
- From Canada, call 1 (800) 345–2384.
- Outside the continental United States, contact your local CONVEX office, or contact the TAC directly at 1 (214) 497–4379.

CXSOFT customers can contact the CXSOFT Technical Assistance Center (TAC) at the following numbers:

- Within the United States and Canada, 1 (800) 426–8979.
- Outside the United States and Canada, 1 (214) 497–3110.

Introduction and usage 1

This chapter explains how to access PRTSMail. Prerequisites are discussed, as well as descriptions of the commands and examples of how to use PRTSMail.

Prerequisites

An identification key is the prerequisite for using PRTSMail. The key can be obtained by using the `-k` option to the `contact` utility.

C Series customers

TAC Patch V10.1.137 is required for a new version of the utility and the man page. This patch may be obtained through your local CONVEX office and works with versions 9.0 through 10.2 of ConvexOS and Utilities.

This patch is not necessary after ConvexOS and Utilities V10.2; it is included in ConvexOS and Utilities V11.0.

Other customers

Customers who purchase support for CXSOFT products receive CXSOFT Contact V1.0 which has the `-k` option.

Using PRTSMail

To use PRTSMail, send electronic mail to *prtsuser* at CONVEX:

- For users of UUCP, the address is `convex!prtsuser`
- For users of the Internet, the address is `prtsuser@convex.com`

The subject line of the mail must be the keyword `PRTS` for PRTSMail to work properly. The PRTSMail server is case insensitive, so `PRTS`, `prts`, and `pRtS` are all functionally equivalent.

The text of the message must contain one command per line. The commands are detailed in the next section, "PRTSMail commands."

PRTSMail commands

The syntax of PRTSMail commands is

command [*option* ...]

where command is the help, key, or status command followed by zero or more *options* modifying that command. Presently, there are only three commands. Multiple commands can be specified in one piece of mail to PRTSMail.

help

The help command sends a help message that explains how to use PRTSMail. Specifying the help command ignores the rest of the command line.

The help message is also sent if the message text is empty, the database query returns no information, no key is specified, or an invalid key is specified. Figure 1 is a request for the help message.

Figure 1
Help message request

```
convex% mail prtsuser@convex.com
Subject: PRTS
key 123456789012
help
^D
convex%
```

Only one copy of the help message is sent in the mail to the requestor. This is to minimize the length of the response mail from the PRTSMail server because some mail gateways truncate mail beyond a certain length. For example, some machines only send the first five hundred lines or 25000 characters and drop the rest.

key

The `key` command is used to identify the sender as a valid CONVEX customer. It must be the first command in the text of the request mail. The syntax is

`key` *number*

where the actual key, *number*, can be obtained by using the `contact -k` command. The key is twelve digits long with no hyphenation or white space.

The `contact -k codeword` command takes the code word you specify and generates an encrypted key for identification. The code word may be obtained by contacting your local CONVEX office. CXSOFT customers should have received the code word in their support package. Contact the CXSOFT TAC as defined in the Preface if you have lost your code word.

When the `key` command reaches the PRTSMail server, it decrypts the key. If the decrypted key is valid, the request is answered and the results returned via electronic mail. If the key is invalid, an error message `Either an invalid key was found or no key was found for this request` is returned with the help message.

An example of using the `contact -k` command is shown in Figure 2.

Figure 2
`contact -k codeword`

```
convex% vers /usr/convex/contact
/usr/convex/contact: 10.1.137
convex% contact -k applesauce
214038151051
convex%

hp9000% vers /usr/convex/contact
/usr/convex/contact:
     Convex contact utility V1.0 for HP-UX
hp9000% contact -k applesauce
246247151051
hp9000%
```

Note

`applesauce` is not the correct code word and the key generated, `214038151051`, is invalid.

status

The syntax of the `status` command is

```
status {pr | bug | cpu | topic} (number [,number,...])
   [open] [date mm/dd/yy]
```

where

- `status` is the keyword used to obtain status information.
- `pr`, `bug`, `cpu`, or `topic` is the keyword specifying the desired information:
  - `pr` is used to request the specified Problem Report (PR) number or numbers.
  - `bug` is used to request the specified bug report (X-record) number or numbers.
  - `cpu` is used to request the specified CPU serial number or numbers. For CXSOFT customers, this is your ten-digit customer number.
  - `topic` is used to request problems in the specified topic number or numbers.
- *number* is the item to search for in the database query.
- `open` is an optional keyword for the CPU serial number search. The topic search only looks for open reports.
- `date` is an optional keyword for the CPU serial number and topic searches. It must be followed by a date in the form *mm/dd/yy*, where
  - *mm* is the month digit (leading zero optional, 1–12).
  - *dd* is the date (leading zero optional, 1–31).
  - *yy* is the year, in either two- or four-digit form.

The help message is returned if the `pr`, `bug`, `cpu`, or `topic` keywords are omitted, the parentheses are omitted, or any other syntax error is made.

At least one number must be specified inside the parentheses—specifying no numbers returns the help message. More than one number may be specified by using commas inside the parentheses. White space (spaces and tabs) is allowed on either side of the commas.

The length of any single command is limited due to internal data structure sizes. However, since multiple commands can be in one piece of electronic mail, this should not be a problem. For the sake of readability, it is recommended that commands be limited to 80 characters in length.

The following figures explain how to obtain the status of PRs.

Figure 3
Obtain the status of a single PR

```
convex% mail prtsuser@convex.com
Subject: PRTS
key 123456789012
status pr (18346)
^D
convex%
```

In Figure 3 the status of a single PR with an index number of 18346 is requested. This example displays the Internet-style address for electronic mail.

Figure 4
Obtain the status of two PRs

```
convex% mail convex\!prtsuser
Subject: PRTS
key 123456789012
status pr (12590, 17860)
^D
convex%
```

In Figure 4 the status of two PRs, with indices of 12590 and 17860, is requested. This example uses the UUCP-style address for electronic mail. Note that the exclamation point must be escaped with the backslash at the C shell command line; the backslash is not required for Bourne shell users. Also note that the example performs the same queries if the one `status` command is turned into two, as shown in Figure 5.

Figure 5
Obtain the status of two PRs with two commands

```
convex% mail convex\!prtsuser
Subject: PRTS
key 123456789012
status pr (12590)
status pr (17860)
^D
convex%
```

Figure 6
Obtain status of all PRs for a CPU

```
convex% mail prtsuser@convex.com
Subject: PRTS
key 123456789012
status cpu (123)
^D
convex%
```

The example in Figure 8 requests the status on every PR in the database that was submitted from CPU number 123.

Figure 7
Obtain status of all open PRs for a CPU

```
convex% mail prtsuser@convex.com
Subject: PRTS
key 123456789012
status cpu (123) open
^D
convex%
```

The example in Figure 8 requests the status of every open problem report submitted against CPU 123 in the database.

Figure 8
Obtain status of all PRs against a specific topic

```
convex% mail prtsuser@convex.com
Subject: PRTS
key 123456789012
status topic (3)
^D
convex%
```

The example in Figure 8 requests the status of every open problem report submitted against topic 3 (the CONVEX Share Scheduler) in the database. For a list of available topic numbers, refer to Appendix A, "Problem report topic numbers."

Figure 9
Obtain status of all PRs against a specific topic since a given date

```
convex% mail prtsuser@convex.com
Subject: PRTS
key 123456789012
status topic (3) date 1/1/93
^D
convex%
```

The example in Figure 8 requests the status of every open problem report submitted since January 1, 1993, against topic 3 (the CONVEX Share Scheduler) in the database. For a list of available topic numbers, refer to Appendix A, "Problem report topic numbers."

Common questions

This section answers some frequently-asked questions about using PRTSMail.

Address formats

How do I know which format address to use?

Check the /usr/lib/contactcap file and look for the `:uu:`, `:pa=:` and `:mb=:` fields. If they look like

```
:uu:pa=convex:mb=contact:
```

then you are probably using UUCP. If they look like

```
:uu:pa=:mb=contact@convex.com:
```

then you are using the Internet. If there is no `:uu:` field, your site is not configured to send Problem Reports via electronic mail.

Identification keys

How do I know what my identification key is?

Use the `contact -k` command to determine your current identification key.

What if I can't get my identification key with the `contact -k` *command, or I need to get my code word?*

Contact your local CONVEX office and ask for TAC Patch V10.1.137. This patch is a new version of the `contact` utility and its man page, and it will work for ConvexOS V9.0 through V10.2, inclusive. Beginning with ConvexOS V11.0, this version of the `contact` utility will be provided automatically. Your local CONVEX office will also supply the code word needed to generate the key.

CXSOFT customers who purchase support from CXSOFT receive a version of the `contact` utility which has the `-k` option. The code word is included in your support package. Contact the CXSOFT TAC if you have lost your code word.

Configuration

Who do I call for help in configuring PRTSMail or `contact`*, or if my PRTSMail request bounces?*

Contact your local CONVEX office or the Technical Assistance Center, as defined in the Preface. The problem could be a poorly-configured /usr/lib/sendmail.cf file, or a problem at CONVEX, such as a down server machine.

Response

How long should it take for me to receive my response?

Most requests receive responses within 48 hours. Due to mail configurations such as multiple UUCP hops or mail gateways, CONVEX cannot guarantee a definite time frame. However, if you do not receive a response within 72 hours, this may indicate a problem with the server or that your mail was lost. Contact your local CONVEX office or the TAC for assistance.

How do I interpret the returned mail from the database query?

Refer to Chapter 2, "Interpreting the response." If that fails to clarify matters, contact your local CONVEX office or the TAC, as defined in the preface.

Interpreting responses 2

This chapter discusses the response mail returned upon querying the PRTS database. The fields returned by a successful query are discussed, and the results from the queries made in Chapter 1, "Introduction and usage," are given.

PRTSMail responses

Once the electronic mail containing the query reaches the PRTSMail server at CONVEX, the server processes the request.

If no `key` command is specified, the error message `Either an invalid key was found or no key was found` is returned with the help message. If the command is `help`, the server returns the help message. If the command is `status`, the server performs a database query for the PR number(s), CPU serial number(s) or customer number(s), or topic number(s) specified inside the parentheses. When the query is completed the information obtained is returned to the sender via electronic mail.

The following sections describe what information is returned from a successful database query. The field names are listed in the order that they appear in the response from PRTSMail.

Problem Report

The Problem Report field contains the Problem Report index number. This is the five-digit number that the PRTS database assigned to the contact report when it was submitted.

CPU

The CPU field contains the CPU serial number of the machine against which the Problem Report was submitted.

For CXSOFT customers the CPU field is your ten-digit customer number.

Received

This field contains the date and time that the Problem Report was received by the PRTS database.

Priority

This field is the assigned priority, defined by the submitter, of the Problem Report and the keyword associated with it. Possible values are listed in Table 1.

Table 1
Priorities

Priority	Description
1	Critical
2	Serious
3	Necessary
4	Annoying
5	Enhancement
6	Informative

Status

This field represents the current status of the Problem Report. Possibilities are:

- **Bug Created**—A bug report has been created and is assigned to CONVEX Engineering.
- **Closed**—The Problem Report (and all associated bugs) has been closed.
- **Under Review**—The TAC is reviewing the Problem Report.

When a bug is created and when the Problem Report is closed, the date is also given in the Status field, in the format *mm/dd/yy*.

Product

The Product field is the product name against which the Problem Report was submitted. This could be general, such as ConvexOS and Utilities, or specific, such as cp.1 man page.

Version

The Version field is the version of the product against which the Problem Report was submitted.

Summary

This field is the one-line summary or brief description of the problem for which `contact` asked the submitter.

Description

The Description field is the detailed description of the problem as phrased by the submitter. This field can include subsections:

- Rpt-By—Instructions on how to reproduce the problem
- Doc—Any documentation comments

if either section was in the PR as submitted.

Only the first 512 bytes of the actual description are stored in this field.

Related Bug

If a bug report has been created against a Problem Report, any of the following five fields can appear. Not all bug reports contain all these fields all of the time. Some Problem Reports can have more than one associated bug report. If this is so, multiple Related Bug sections appear, one per bug report.

Bug Status

This is the status of the bug report. Possible values for this field are:

- Bug—The problem is a bug.
- Enhancement Request—This is a request for new functionality.

Bug Disposition

The current disposition of the bug can be one of eight possible choices. These are:

- **Closed**—The enhancement request has been closed as impossible to grant.
- **Duplicate**—This was a duplicate bug report and has been merged with another bug report for CONVEX Engineering to resolve. Note that the Problem Report remains open until all associated bugs are resolved.
- **Fixed**—The bug has been fixed or the enhancement request has been granted, and the bug report has been closed.
- **Not Reproducible**—The problem could not be reproduced by CONVEX Engineering. The bug report has been closed.
- **Open**—The bug report is currently open and under investigation by CONVEX Engineering.
- **Pending Closure**—The bug has been fixed, but the bug report has not been closed. This is usually because one group within CONVEX Engineering has to close out their assignment to the bug report. For example, the problem has been fixed but a test case to ensure against regressions has not been written.
- **Restriction**—The problem is a restriction, and the bug report has been closed. The usual restrictions are trying to use a utility for a purpose beyond its design specifications, such as expecting the `ls` command to back up a file system.
- **User Error**—The problem is not a bug, but an error on the part of the user, such as using incorrect syntax or nonexistent options. The bug has been closed.

For all dispositions except **Open**, the date of closure is given, in the format *mm/dd/yy*. The date the bug was opened (created) is given in the Problem Report **Status** field for open bug reports.

Bug Description

This is the description of the associated bug, described with enough details for CONVEX Engineering to resolve the problem.

If the description is unavailable it means that the text has not been checked for accuracy and precision, and that nobody in the Engineering or TAC groups has proofread the description.

Bug Resolution

The resolution, or actions taken by the engineer(s), in resolving the problem or answering the request. If the problem is not resolved, this field will be empty.

The resolution can be unavailable for two reasons:

- The bug report is not yet closed and there is no resolution.
- The resolution has not been proofread by a member of the Engineering or TAC organizations.

Version resolved in

This is the version of the product in which the bug is resolved. If the problem is not resolved, this field is not present.

Message format

The format of the response mail from PRTSMail is illustrated in Figure 10.

Figure 10
`status pr (27116)` **output**[1]

```
①  Problem Report: 27116   ②CPU: 8202

③  Received: Mon Jul 6 21:57:00 1992

④  Priority: 5 (Enhancement)  ⑤Status: (Closed on 7/20/92)
⑥  Product:
   CONVEX Press Catalog

⑦  Version: 1.0

⑧  Summary:
   Catalog does not cover hardware manuals.

⑨  Description:
   The CONVEX Press Catalog, distributed at the most recent User Group,
   lists no hardware manuals. (Only the DSW-xxx series is included; the
   DHW-xxx series is nowhere to be seen.) It would be useful if the
   hardware manuals were also listed, either in this catalog or in a
   "hardware documents" catalog.

   Doc:
   See above. This CONVEX Press Catalog _is_ a very useful little
   booklet. I would hope that you would continue to update and release
   it on a regular basis.
⑩  ---------------- Related Bug:

   Bug Status: Bug
   Bug Disposition: Fixed on 7/17/92
   Bug Description:
   The CONVEX Press Catalog lists no hardware manuals. It would be useful if
   the hardware manuals were also listed, either in this catalog or in a
   "hardware documents" catalog.

   Bug Resolution:
   The CONVEX Press catalog is being revised. As part of this effort, we will
   look at including hardware manuals.  However, because this catalog is
   targeted at customers, and many of our hardware manuals are internal
   documents and not available for direct order for customers, not all hardware
   manuals will be listed in the catalog.

   Version resolved in: 1.1
```

[1] The output from the `status pr (27116)` command has been modified to show the correct version number of the product in question.

The numbers inside circles represent the fields discussed in the "PRTSMail responses" section of this chapter.

① PR is assigned an index number of 27116.

② PR was submitted from CPU serial number 8202.

③ The PRTS database received the PR on Monday, July 6, 1992, at 9:57 pm Central Time.

④ PR was submitted with a priority of `5 (Enhancement)`, an enhancement request..

⑤ PR was closed on July 20, 1992.

⑥ PR was submitted against the *CONVEX Press Catalog*.

⑦ The version number was 1.0, the First Edition of the document.

⑧ The summary line describes the problem: that the catalog did not contain information about hardware documentation.

⑨ The description explains the summary line. Note that the Doc section is included, as the original PR had a Documentation Comments section attached.

⑩ The related bug report (created as a bug because the enhancement request indicated a problem with the catalog) was marked `Fixed on 7/17/92`. (Note that the PR was not closed until July 20th.) The description of the bug report is a more precise statement of the enhancement request, asking that the hardware manuals be included in the documentation catalog. The resolution of the bug report is the response from the Documentation group. The `Version resolved in` field is 1.1, for the first revision to the first edition of the document.

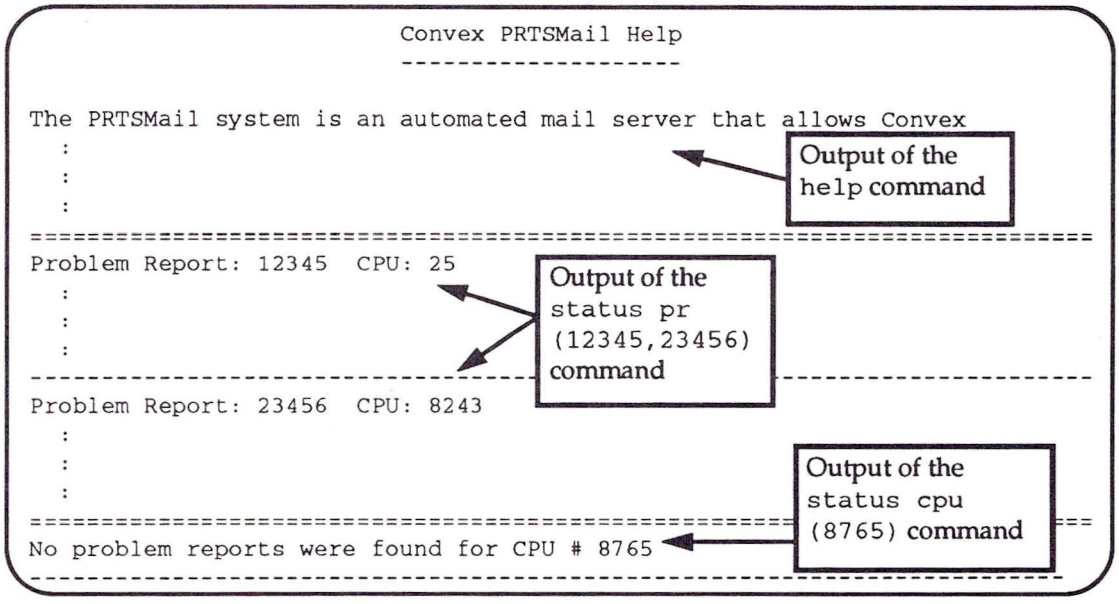
Message separators

Each message into the PRTSMail server results in one response message back, regardless of the number of commands in the original message. If a request comes into the PRTSMail server for exactly one Problem Report (PR), the response mail contains just the one PR. If one command line brings up more than one PR, the individual PRs are separated by a line of minus signs. Multiple commands are separated by a line of equal signs. Figure 11 and Figure 12 illustrate this.

Figure 11
Customer sends mail to the PRTSMail server

```
convex% mail prtsuser@convex.com
Subject: PRTS
key 123456789012
help
status pr (12345,23456)
status cpu (8765)
^D
convex%
```

Figure 12
Customer receives response mail



The response to the one electronic mail message sent from the customer to the PRTSMail server is one electronic mail message containing the output of the commands in the body of the request. Since the `help` command came first, the first segment of output (delimited by the equal signs) is the help message. The second command asks for two PRs by number, so the next segment of the output is the first PR status, a line of minus signs, and the second PR status. Finally, the request asked for all PRs submitted from CPU number 8765, so the final segment of output is each of those PRs, each separated with a line of minus signs.

Examples of responses

The responses to the sample queries issued in Figures 1 through 9 in Chapter 1, "Introduction and usage," are given in this section. This allows you to compare the mail you sent with the response you receive.

Figure 13
The `help` message

```
                        Convex PRTSMail Help
                        --------------------

The PRTSMail system is an automated mail server that allows Convex
customers to review the status of problem reports (PRs) that have been
submitted to Convex via the contact(1) facility.

To receive status on a problem report send a mail message to

     "prtsuser@convex.com"

with the subject of the mail message: "prts"

The syntax for using this service is one "command" per line in the form of:

   <command> [option ...]

There are three basic commands for this service.  The commands should be
placed in the body of the mail message.  Use only one command per line.
The commands are:

   1) "key"          Identify the sender as a Convex customer.  This must
                     be the first command in the message.  For example:

                             key 123456789012

   2) "help"         This message.

   3) "status"       Retrieve the status of these problem reports, or
                     the status of problem reports submitted from these
                     CPU serial numbers, or the status of problem reports
                     for this topic.  Examples follow.

To tell the PRTSMail server to retrieve the status of problem report #18346
for example, use the following command in the body of the mail message:

   status pr (18346)

To tell the PRTSMail server to retrieve the status of more than one problem
report, separate the problem report numbers by commas.  Example:

   status pr (12590, 17860)

This would retrieve the status of these two problem reports.

To tell the PRTSMail server to retrieve bug report #6852, use the following
command:

   status bug (6852)
```

Continued on next page...

Figure 13 (continued)
The help message

```
To tell the PRTSMail server to retrieve the status of all problem reports
submitted from a particular CPU number use the keyword "cpu" with the "status"
command.  For example:

   status cpu (8246)

This would retrieve the status of all the problem reports submitted from
CPU serial number 8246.

By including the keyword "open" after a 'status cpu' command, only open
problem reports for the specified cpu will be retrieved.  For example:

   status cpu (8246) open

This would retrieve the status of all the open problem reports submitted
from CPU serial number 8246.


To tell the PRTSMail server to retrieve the status of all problem reports
for a specific topic, use the "topic" keyword and topic number with the
"status" command.  For example:

   status topic (3)

This would retrive the status of all the open problem reports that have been
reported against the Convex Share Scheduler (topic code 3).
See below for a list of topic codes and the topic name.

To tell the PRTSMail server to retrieve the status of problem reports that
were received after a certain date, append the keyword "date" after the
status command.  For example:

   status topic (3) date 1/1/1993

This would retrive the status of all the open problem reports that have been
reported against the Convex Share Scheduler (topic code 3) since January 1,
1993.


The turn-around time for most requests should be within 48 hours.  However,
please allow up to 72 hours in case of mail delays.  If you have problems
with this subscription system, send mail to:

                    prtsmail-help@convex.com
```

A topic list would be included here; refer to Appendix A.

Figure 14
`status pr (18346)` output[2]

```
Problem Report: 18346   CPU: 8243

Received: Thu Mar 14 08:50:00 1991

Priority: 3 (Necessary)     Status: (Closed on 3/20/92)
Product:
fsck

Version:  9.0

Summary:
fsck does not deal with files which are the maximum allowed sizemaximum

Description:
fsck does not handle files 2^31-1 in size well.  When convex4 crashed
this morning, fsck complained that it couldn't correct problems with
a file I had created which consisted of a 16k block at offset 0 and
another 16k block at offset 2^31-16k (with a hole for the rest of the
data).  After running fsck manually, it removed the file and went on
its merry way.

Rpt-by:
create a file 2G-1 in size and then fsck the partition.  I don't know
if this only happens on files with holes or if it happens with any file


---------------- Related Bug:

Bug Status: Bug
Bug Disposition: Fixed on 10/21/91
Bug Description:

fsck does not handle files 2^31-1 in size well. When rebooting fsck
complained that it couldn't correct problems with a file which
consisted of a 16k block at offset 0 and another 16k block at offset
2^31-16k. After running fsck manually, it removed the file and
continued on correctly.

Bug Resolution unavailable.
Version resolved in: 10.0
```

[2] This example has been modified to correct typographical errors and change a customer's host name.

Figure 15
`status pr (12590,17860)` output[3]

```
Problem Report: 12590   CPU: 147

Received: Tue Feb  6 13:33:00 1990

Priority: 3 (Necessary)     Status: (Closed on 3/13/91)
Product:
UNIX Utilities / newst

Version:  8.0

Summary:
<NONE>

Description:
Newst does not allow you to change the percentage of minimum free space
threshold (minfree), as does newfs. Newst should allow you to do this,
as a great deal of space can be wasted with large stripes.
---------------- Related Bug:
Bug Status: Bug
Bug Disposition: Fixed on 3/13/91
Bug Description:
newst does not allow the user to change the percentage of minimun free space
threshhold (minfree), as does newfs. newst should allow the user to do this,
as a great deal of space can be wasted with large stripes.
Bug Resolution unavailable.
Version resolved in: 10.0
--------------------------------------------------------------------------------
Problem Report: 17860   CPU: 8438

Received: Fri Feb 22 15:35:00 1991

Priority: 4 (Annoying)     Status: (Closed on 9/5/91)
Product:
/etc./newst

Version:  8.0

Summary:
want -m flag

Description:
We recently put together a large stripe work partition consisting
of almost 2 gigabytes of storage.  My job was trying to use it
to sort 1.8GB of data, but ran out of space when usage reached
91%.  Under the newfs command there is a -m flag to change the
default 10% of disk space reserved for root only.  In this case
I really needed the extra space and there was no reason to reserve
it for root.
---------------- Related Bug:
Bug Status: Wish
Bug Disposition: Fixed on 8/5/91
Bug Description:
The -m flag of newfs allows the modification of reserved space on a file
system. This flag would be quite useful in the newst command.
Bug Resolution unavailable.
Version resolved in: 10.0
```

[3] This example has been modified slightly to correct typographical errors.

Figure 16
`status cpu (123)` output

```
Problem Report: 3785    CPU: 123

Received: Tue Mar 17 13:06:00 1987

Priority: 2 (Serious)     Status: (Closed on 7/10/90)
Product:
fc distribution tape, v2.2

Version:  2.2

Summary:
McAir's FC 2.2 tape had 128 as both primary and secondary serial #'s

Description:
The FC 2.2 tape we sent to McAir had "128/123" marked as the serial
numbers on the sticky label, but the compiler refused to run on
CPU 123 (both serial numbers in the compiler were 128).

I was able to login to their system and patch their compiler (this
will not be possible in future release of FC).
----------------- Related Bug:
Bug Status: Bug
Bug Disposition: Closed on 7/10/90
Bug Description:
The FC 2.2 tape we sent to McAir had "128/123" marked as the serial
numbers on the sticky label, but the compiler refused to run on
CPU 123 (both serial numbers in the compiler were 128).
I was able to login to their system and patch their compiler (this
will not be possible in future release of FC).

Bug Resolution unavailable.

--------------------------------------------------------------------------------
Problem Report: 14243    CPU: 123

Received: Tue Jul 24 17:09:00 1990

Priority: 2 (Serious)     Status: (Bug Created on 7/24/90)
Product:
UNIX Utilities / crashdump

Version:  8.1

Summary:
<NONE>

Description:
running crashdump over ethernet gives the following error:

crashdump:  CCU 7: Ethernet configuration timeout.
----------------- Related Bug:
Bug Status: Bug
Bug Disposition: Closed on 6/13/91
Bug Description unavailable.
Bug Resolution unavailable.
```

Additional problem reports deleted

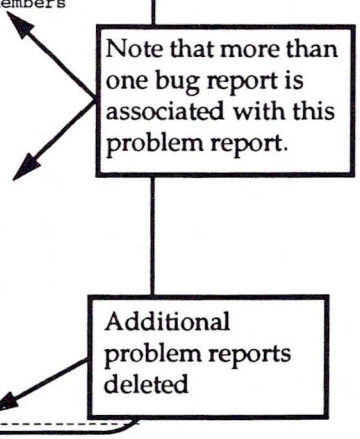
Figure 17
`status cpu (123) open` output

```
Problem Report: 14243   CPU: 123

Received: Tue Jul 24 17:09:00 1990

Priority: 2 (Serious)     Status: (Bug Created on 7/24/90)
Product:
UNIX Utilities / crashdump

Version:  8.1

Summary:
<NONE>

Description:
running crashdump over ethernet gives the following error:

crashdump:  CCU 7: Ethernet configuration timeout.

----------------- Related Bug:

Bug Status: Bug
Bug Disposition: Closed on 6/13/91
Bug Description unavailable.
Bug Resolution unavailable.

------------------------------------------------------------
```

Additional problem reports deleted

Figure 18
`status topic (3)` output

```
Problem Report: 12927   CPU: 16406

Received: Tue Mar  6 15:48:00 1990

Priority: 2 (Serious)     Status: (Bug Created on 8/30/90)
Product:
sharecf

Version:  8.0

Summary:
The "gname" construct only examines /etc/group, not /etc/passwd

Description:
According to the documentation for "sharecf":\\      If "gname" or "gid" is speci
fied, "groupname" or "gid" is expanded      in place to the list of members in t
he group.\\However, only the /etc/group file is examined to determine the "membe
rs inhe group".  The /etc/passwd file isn't consulted at all.  Thus, you can\n
ot use the "gname" or "gid" constructs to place users in a scheduling,roup base
d on their login group, unless you've gone to the extra (and\otherwise unnecessa
ry) expense of adding their logi

---------- Related Bug:

Bug Status: Enhancement Request
Bug Disposition: Open Bug
Bug Description:

sharecf should display message (issue warning) when group with no members
is used.

Bug Resolution unavailable.


---------- Related Bug:

Bug Status: Enhancement Request
Bug Disposition: Open Bug
Bug Description:

The documentation of the "gname" and "gid" constructs in the manual
page are too terse and ambiguous.  In addition, it is not clearly
stated that these constructs can not appear as part of a "member"
construct.

Bug Resolution unavailable.
```



Note that in Figure 18, since the bug reports are not closed the resolutions are unavailable.

Figure 19
`status topic (3) date 1/1/93` output[4]

```
Problem Report: 29891   CPU: 28750

Received: Sat Jan  9 12:01:00 1993

Priority: 2 (Serious)     Status: (Bug Created on 1/21/93)
Product:
Share Scheduler

Version:  10.0

Summary:
gname= and gid= identifiers do not work properly.

Description:
Neither the gname= nor the gid= identifiers look at /etc/passwd file. They
only detect users listed in /etc/group.

Rpt-by:
Create a share config file with entry of "gname=foo shares=100;" and load it.

----------------- Related Bug:

Bug Status: Bug
Bug Disposition: Open Bug
Bug Description unavailable.
Bug Resolution unavailable.


--------------------------------------------------------------------------------
```

[4] This example has been modified slightly to correct typographical errors.

Problem report topic numbers

A

This appendix contains the topic numbers for the `status topic` command.

Table 2
Problem report topic numbers

Topic number	Description
1	68000 Tools
2	`pcc` (Portable C Compiler)
3	CONVEX Share Scheduler
4	Consultant (`csd`)
5	COVUEshell
8	Device Drivers
9	`fc` (FORTRAN Compiler)
15	TCP/IP (CONVEX Internet Services)
16	Runtime Libraries
17	SPU OS
19	ConvexOS/Secure
20	ConvexOS Utilities
22	`cc` (CONVEX C Compiler)
23	VECLIB
24	Ada
25	NFS (Network File System)

Table 2 (continued)
Problem report topic numbers

Topic number	Description
27	COVUEnet
28	COVUEedt
29	COVUEbatch
31	COVUElib
35	ConvexOS I/O
36	ConvexOS Kernel
38	CXwindows
39	CXbatch
40	NCS (Network Computing System)
41	CXpa (CONVEX Performance Analyzer)
43	UltraNet
45	COVUEbinary
46	CONVEX Toolbox
48	C++
49	CXdb (CONVEX X-based Debugger)
50	`lint`
51	Motif
52	PHIGS
53	X Terminals
56	CONVEX Visual Toolkit
57	Application Compiler
60	OSI WAN Transport
61	OSI LAN Transport
62	OSI FTAM
63	OSI X.400 MHS

Table 2 (continued)
Problem report topic numbers

Topic number	Description
65	Math Advantage
66	GIP
68	CSM (CONVEX Storage Manager)
69	CXmetrics
76	PEX
77	HIPPI
78	FDDI
79	UXE (UNIX Environment)
80	RTK (Real Time Kernel)
82	ALL (as, ld, libs)
83	Unitree
84	ITC (Integrated Tape Controller)
94	Interlanguage Programming Guide (ILPG)
96	OSI Transport Services
97	OSI X.25 Connection
98	OSI Ethernet Connection
99	CONVEX Tape System
115	EMASS FileServ
116	CXaims
117	ConvexPVMdb

Index

ORDER NUMBER
DSW-600

DOCUMENT NUMBER
700-044030-004