FUJITSU COMPUTERS SIEMENS

Reliant UNIX ONLINE Documentation

Reliant UNIX 5.45

Diagnostics and Troubleshooting

Edition October 2000

Identification: U42309-J-Z915-2-76

Copyright and Trademarks

1 Preface

The computers in the RM series are extremely reliable systems which are made with the latest manufacturing processes and are being enhanced continually. Nevertheless, the current state of the art cannot preclude the possibility of individual hardware components and subsystems failing. Furthermore, external influences, such as power failure or incorrect handling, and software errors can result in unexpected situations (hang-ups, system crashes etc.). It is therefore necessary to make a reasonable estimate of potential problem scenarios in order to establish an approach to error management and thus, at least, minimize the impact of any errors that may occur and allow troubleshooting to take place without affecting system availability.

The RM diagnostic system is part of a high availability strategy under Reliant UNIX ® in an attempt to improve availability, diagnosability and serviceability by a combination of monitoring, redundancy and preventive action. These efforts are subsumed under the abbreviation RAS (Reliability/Availability/Serviceability).

Preventive action aims to avoid all malfunctioning of the system as far as possible. This includes particular hardware design measures ranging from simple parity checks in memory, through greater integration in fewer functional units, to development of component assemblies which can be replaced while the system is running. As far as software engineering is concerned, efforts are focused on developing high-quality products that avoid errors and alleviate the consequences of them. These products are intended to behave predictably when unexpected events occur (e.g. power failure).

Financial constraints mean that not all possible preventive action has been taken to increase system availability so the RM diagnostic system incorporates error handling routines, for instance automatic correction of recurrent or correctable malfunctions (e.g. of CPU cache, memory, hard disks) and logging of hardware faults or incorrect behavior of system and application software.

1.1 Arrangement of the manual

This manual consists of three logical parts.

Part 1 (chapters 2-3) describes errors that can occur while the system is starting up and while it is running, how the computer responds to these errors and what action you can take to deal with and possibly eliminate them. This description is split up according to the computer models.

Part 2 (chapters 4-11) describes the user interfaces that are available for system diagnostics. These are:

- Evaluation and administration of logbooks
- Evaluation of system device logs
- Hardware diagnostics under VConfig and Config
- Provision of error information in repair cases
- System diagnostics for Reliant UNIX systems
- Test and diagnostic system for checking the hardware
- Brief description of ServerView for RM200/300/400 systems.
- Additional diagnostic resources

Part 3 (chapters 12-17) contains

- Descriptions of the following routines:
  Switching to the board debug monitor (RM600)
  Switching to the PROM or firmware monitor (RM200/300/400)
  Loading the Mini System
  Resolving inconsistencies in the NVRAM
  Loading the old system kernel
  How to boot the system when the system disk has failed
  Powering up in single-user mode
- Information on errors that can occur in conjunction with the cabling and startup of peripherals.
- A description of the effects of a power failure in system configurations with and without a BBU, UPS or redundant power pack.
- An explanation of the LAR concept
- A list of error messages in the boot1 phase on RM600 systems
- A short description of how to find information on error messages relating to system calls

1.2 Target groups

This manual is primarily targeted at system administrators and Service staff who are acquainted with system diagnostics. It is also aimed at programmers who, for example, want to use the Logging application programming interface (API) in their own applications.

Good knowledge of the operating system and hardware is required, as is familiarity with character-oriented user interfaces that have been implemented with FMLI (Form and Menu Language Interpreter) and with the graphical user interfaces WebSysAdmin and SINIX/windows.

1.3 Notational conventions

The following notational conventions are used in this manual:

italics Names of commands, system calls, functions, files, procedures, programs etc., and menu items, input fields and output fields of figures in continuous text

<italics>
Variables in continuous text, system outputs, and examples

name(extension)
Commands, system calls, functions, files, procedures etc. for which manual pages (descriptions) exist are shown with an extension in parentheses. The extension indicates the chapter containing the manual page in one of the following books:

- User's Reference Manual, see also *ref_urm*(1)
- System Administrator's Reference Manual, see also *ref_sarm*(1)
- Programmer's Reference Manual, see also *ref_prm*(1)
- Networking Reference Manual, see also *ref_net*(1)

You can also view manual pages on screen with the command *man* (see *man*(1)). You can find out more about this command by viewing its manual page. Enter *man man* to do so. Refer to manual page *intro*(1) for details of printing a manual page.

typewriter text
: System output such as: error messages, other messages, notes, file extracts and sample programs

bold typewriter text
: User input in examples

"quotes"
: References to other chapters or manuals

▶ □□□□□□User actions

 Additional information that should be observed to understand passages nearby

 Warnings that must be observed

1.4 Telephone and fax numbers of the support center

Germany

Siemens IT Service GmbH, Call Management Center (CMC)

Telephone number:	(01 80) 5 40 40
Fax number:	(01 80) 5 33 67 99

USA and Canada

Siemens Corporate Customer Support Center (CCSC)

Telephone number USA and Canada:	(800) 6 95-22 72
Telephone number international:	(408) 4 28-90 00
Fax number (primary):	(408) 4 28-74 50
Fax number (secondary):	(408) 4 28-70 20

International (except Germany, USA and Canada)

Ask your local Siemens AG provider or branch for the telephone and fax numbers of a support center nearby.

2 RM600 troubleshooting

This chapter is intended to guide you through steps in handling various error situations. This will enable you to get your computer running properly again in the event of minor errors, and to prepare diagnostic documentation for analysis by Service in the event of serious errors.

This chapter is split into two sections.

The "System startup" section explains how you can diagnose errors that occur while the computer is powering up, and resolve them if possible.

The "System running" section tells you how to diagnose problems that occur while your computer is up and running, and how to solve them if possible. It also explains how to prepare the relevant diagnostic documentation.

What else you need to consider:

Have your system ID number (SY...) at hand when reporting problems to Service. It is shown on a gray sticker on the system unit.

Heed the safety instructions in the operating manual for the individual devices (monitor, system unit, auxiliary cabinet, terminal, printer, UPS etc.).

Experience has shown that it is always worth taking a look at the operating and installation instructions for your system if an error occurs. The device- and system-specific advice on troubleshooting that is given in those documents will also be of assistance in solving problems.

2.1 System startup

This section describes errors that can occur during system startup, how your computer will respond to them as standard, and what action you can take to deal with them.

System startup is logged on the console. Console messages can often be seen for just a short time because they are scrolled on the screen. To make sure you don't miss any information, you can halt screen output by pressing [CTRL][S] and resume it with [CTRL][Q]. Important console messages during the boot phase, especially kernel and driver messages, are logged in Logging □V3.0.

2.1.1 Turning on the system

The following problems can occur when the system is turned on:

2.1.1.1 RM600 E

- The *ON/BATT* LED on a cabinet does not light up and the LCD does not display anything.

  No power is being supplied or the system was not turned on properly.
  - Check the power supply or turn the system on again. Notify Service if the system still cannot be turned on.

- The *ON/BATT* LED on a cabinet is green but the LCD does not display anything.

  The LCD is defective.
  - Turn the system on again and notify Service even if booting proceeds correctly.

- When the system has been turned on, the LCD briefly displays SYSTEM RESET.

  Internal system parameters could not be reset.
  - Turn the system off and back on again. If SYSTEM RESET appears again, turn the system off and notify Service.

2.1.1.2 RM600-xxx

- The *ON/BATT* LED on the system cabinet does not light up and the LCD on the system cabinet does not display anything.

  No power is being supplied or the system was not turned on properly.
  - Check the power supply or turn the system on again. Notify Service if the system still cannot be turned on.

- The *ON/BATT* LED on another cabinet does not light up.

  No power is being supplied.
  - Check the power supply. Notify Service if the problem is not caused by the power supply.

- The *ON/BATT* LED on the system cabinet is green but the LCD on the system cabinet does not display anything.

  The LCD is defective.
  - Turn the system on again and notify Service even if booting proceeds correctly.

- When the system has been turned on, the LCD on the system cabinet briefly displays SYSTEM RESET.

  Internal system parameters could not be reset.
  - Turn the system off and back on again. If SYSTEM RESET appears again, turn the system off and notify Service.

2.1.1.3 RM600 E and RM600-xxx

- The screen remains dark after turning on the system.

  The screen has not been turned on.
  - Turn the screen on.

  The screen has been dimmed automatically.
  - Press the space bar or the CR key (on 97801 terminals).

  The screen's brightness control is set to dark.
  - Set the brightness control to light.

  The screen has no connection to the system cabinet, or no power is being supplied.
  - Turn the system off and check whether the plug for the power cable and the plug for the cable to the system cabinet have been inserted properly in their sockets. Turn the system back on. Notify Service if the screen stays dark.

- The screen does not show any characters, or just garbled ones.

  The screen parameters have not been set correctly.
  - Set the screen parameters as required. (see the operating instructions for the particular screen or for your computer)

- If there is a malfunction concerning the uninterruptible power supply cabinet, consult the operating instructions for the UPS. These are shipped with the product.

2.1.2 System environment monitoring

During startup, the system environment is monitored by EIPs which are started when the system is turned on and keep track of the power-up process of the computer, the ambient temperature, the functioning of fans and the BBU. With the exception of the UPS cabinet each cabinet has its own EIP. EIPs in the expansion cabinets (RM600 E), MBII and peripheral cabinets (RM600-xxx) and BBU cabinets report malfunctions to the EIP in the system cabinet which is responsible for passing on malfunction reports and reacting to malfunctions.

2.1.2.1 Noncritical situations

If a malfunction occurs which does not jeopardize the system or any part of it (e.g. failure of a redundant fan), then system startup continues and the malfunction is reported as follows:

- On an RM600□E, the EIP reports the malfunction to the HIOS controller in slot 8 or the EHIOS controller in slot 10 (RM600-E30) or 14 (RM600-E70) of the system cabinet. Malfunctions of the UPS are detected by the EIP in the cabinet to which the USP is connected via a contact interface.
- On an RM600-xxx, the EIP reports the malfunction to the CSI controller. Malfunctions of the UPS are detected by the CSI controller directly.

When the operating system has started, the (E)HIOS or CSI controller reports malfunctions to the UBFT driver which in turn notifies the UBFT daemon (see also *ubft*(7) and *ups*(1M)).

Any malfunctions which arise are reported as follows:

- On an RM600-xxx, malfunctions which arise before the UBFT daemon has been started are reported on the console (e.g.):
  ```
  *** WARNING:□
  redundant FAN FAIL in cabinet 3
  ```
- After the UBFT daemon has been started they are
  - reported on the console (e.g.):
    ```
    DRV: Warn: ubft 3042 (48,0) (daemon) cab 0/18: back-lower-middle fan □
    ventilation unit single failure!
    ```
    When Reliant UNIX is running, you can issue the *getemm*(1M) command to see a description of the problem and handling instructions to solve it.
  - and written to the */var/adm/ubft/log* file.

- After the Logging daemon has been started they are
  - recorded by Logging V3.0 (...).
  - entered in the HW_CONFIG database (...).
  - displayed under VConfig (...)

- Remedy the malfunction or notify Service so that defective components can be replaced.

2.1.2.2 Critical situations before the UBFT daemon has started

If malfunctions occur which jeopardize the system or parts of it (e. g. total fan failure) or if Reliant UNIX cannot be loaded within 15 minutes (automatic OFF delay), then the EIP automatically turns off the system or the cabinet in question. If there are problems with the temperature, or if all the fans fail simultaneously then the system will switch off automatically after a few minutes.

The automatic OFF delay timers are started when the system is turned on. The timer counts down to zero in seconds. The current figure is indicated on the system cabinet LCD. The timer is a safety measure in case booting cannot proceed properly because of a malfunction. It ensures that the system does not stay in an undefined state. You can halt the timer by pressing the *MENU* button on the system cabinet.

What happens with malfunctions:

- They are indicated by the LCD and/or LED on the cabinet in question (...).
- They are reported on the console, e. g.:
  - RM600□E:

```
*** WARNING in cabinet 0 (EIP3_ID 32): TEMP Fail□
□
□□□□□□□□□□□□□□□Type in '1' ignore errors and load system□
□□□□□□□□□□□□□□□□□□□□□□□□□□'2' activate BDM□
□□□□□□□□□□□□□□□□□□□□□□□□□□'3' poweroff system□
□
Your choice:
```

  - RM600-xxx:

```
*** WARNING:□
Temperature FAIL in cabinet 3□
□
System possibly NOT bootable!□
□
*** Returning to Monitor□
□
*** Auto Power Off not stopped□
EIP will switch off system!□
□
BDM(MBII 0 t)>
```

System startup is terminated at this point. Your response here depends on the malfunction that has occurred:

- No response is required of you in the event of a power failure or power pack fault as the system is turned off automatically in these cases. If there has been a power failure, turn the system back on when the power failure is over. Notify Service if there is a power pack fault.
- If temperature problems occur, adjust the room temperature or make sure that air can circulate around the ventilation ducts. Refer to the operating instructions for your system to see the required ambient conditions.

  If the malfunction can be eliminated in the specified time, continue with the system startup:
  BDM(SPB□□8□t)>**b**□□□□/*□RM600□E□*/□
  BDM(MBII□□0□t)>**b**□□□/*□RM600-xxx□*/

  or power up the system again:
  BDM(SPB□□8□t)>**creset**□□□□/*□RM600□E□*/□
  BDM(MBII□□0□t)>**creset**□□□/*□RM600-xxx□*/

  If the malfunction cannot be eliminated in the specified time, the EIP turns the system off automatically. Turn the system back on once the malfunction has been remedied. If temperature problems are reported again, turn the system off and notify Service.
- In the event of a total fan failure, turn the system off and back on again. If a total fan failure is reported again, turn the system off and notify Service.

2.1.2.3 Critical situations after the UBFT daemon has started

If a malfunction occurs which does jeopardize the system or some part of it (e.g. total fan failure), it is reported as follows:

- On an RM600□E, the EIP reports the malfunction to the HIOS controller in slot 8 or the EHIOS controller in slot 10 (RM600-E30) or 14 (RM600-E70) of the system cabinet. Malfunctions of the UPS are detected by the EIP in the cabinet to which the UPS is connected via a contact interface.
- On an RM600-xxx, the EIP reports such malfunctions to the CSI controller. Malfunctions of the UPS are detected by the CSI controller directly.

The (E)HIOS or CSI controller reports malfunctions to the UBFT driver which in turn notifies the UBFT daemon (see also *ubft*(7) and *ups*(1M)).

What happens with malfunctions:

- They are reported on the console, e. g.:
  DRV: Err1: ubft 3060 (48,0) (daemon) cab 5: non-redundant fan ventilation failure!
- They are handled by the UBFT daemon as defined by the parameters in the *_upstab*(4) file which you can set to meet the requirements of your system (the default is for the UBFT daemon to shut the system down).
- They are written to the */var/adm/ubft/log* file.
- They are logged in Logging□V3.0 (...).
- They are indicated by the LCD and/or LED on the cabinet (...).

▶ Remedy the malfunction or notify Service so that defective components can be replaced.


You can find further details on system environment monitoring and the functioning of the UBFT daemon in the "Power failure or hazardous temperatures" section of the "Reliant UNIX Operation" manual.

2.1.2.4 LCDs/LEDs

RM600 E

- The *ON/BATT* LED on the system cabinet or on an expansion cabinet is red (with BBU operation only).
  The BBU of the cabinet is active due to a power failure or power pack fault.
- The *ON/BATT* LED on a BBU cabinet is red.
  Power failure or fault in the power pack of the BBU cabinet.
- The LCD on a cabinet indicates TEMP FAIL.
  The room temperature is too high or the cabinet's ventilation ducts are obstructed.
- The LCD on a cabinet indicates FAN FAIL.
  There has been a total failure of the fans.

RM600-xxx

- The *ON/BATT* LED on the system cabinet or the *ON/BATT* and *FAIL* LEDs on an expansion or peripheral cabinet is/are red (only with cabinets of the 2xx, 3xx and 4xx models with an internal BBU).
  The internal BBU of the cabinet is active due to a power failure or power pack fault.
- The *ON/BATT* and *FAIL* LEDs on the BBU cabinet are red.
  The BBU is active due to a power failure or a fault in the power pack of the system cabinet, an expansion cabinet or a peripheral cabinet.
- The LCD on the system cabinet indicates TEMP FAIL and/or the *FAIL* LED on another cabinet is red.
  The room temperature is too high or the cabinet's ventilation ducts are obstructed.
- The LCD on the system cabinet indicates FAN FAIL and/or the *FAIL* LED on another cabinet is red.
  There has been a total failure of the fans.

2.1.2.5 Displaying the cabinet configuration

Before the boot1 phase starts, you see, among other things, information on the console about the status of every cabinet and the version and release of the EIP firmware, e. g.:

RM600 E:

```
Cabinet□□Id□□Type□□□□Status□□□□Version□□□□BBU□□□□□BBU_Stat□
-----------------------------------------------------------□
□□00□□□□□00□□□32□□□□00000000□□□4.0.00□□□□□none□□□□□□----□
□□01□□□□□01□□□33□□□□00000000□□□4.0.00□□□□□none□□□□□□----□
□□02□□□□□02□□□33□□□□00000000□□□4.0.00□□□□□none□□□□□□----□
□□03□□□□□03□□□34□□□□00000000□□□4.0.00□□□□□none□□□□□□----□
□
System:□*□□□□□□RM600□E20
```

RM600-xxx:

```
System Cabinet Configuration:□
Cabinet□□□Status□□□□Version□□□□Release□
---------------------------------------□
□□0□□□□□□□□40□□□□□□□□□□30□□□□□□□□□1□
□□2□□□□□□□□40□□□□□□□□□□30□□□□□□□□□1□
□□3□□□□□□□□40□□□□□□□□□□30□□□□□□□□□4□
□□4□□□□□□□□40□□□□□□□□□□30□□□□□□□□□1□
□
UPS installed: none
```

Theoretically, the status can be represented by values in the range 00 to FF. This is calculated from the bit setting of a byte which is split into four low-order and four high-order bits. The table below shows the status/meaning when the bit marked X is set to 1:

Bit setting	Status	Meaning RM600 E	Meaning RM600-xxx
0000|0000	00	Cabinet turned on, status OK	-
0000|000X	01	Secondary voltage failure	Primary voltage failure
0000|00X0	02	Primary voltage failure	Ambient temperature too high
0000|0X00	04	Non-redundant fan failure	Non-redundant fan failure
0000|X000	08	Critical situation: □ Ambient temperature too high	Redundant fan failure
000X|0000	10	Warning: □ Ambient temperature too high	Cabinet turned off - no secondary voltage
0X00|0000	40	-	Cabinet turned on
X000|0000	80	Cabinet turned off	EIP cannot be addressed or not available

Combinations are possible if various states occur simultaneously, e. g.:

Bit setting	Status	Meaning RM600 E
000X|0X00	14	Non-redundant fan failure and ambient temperature too high

2.1.3 Activating the boards

2.1.3.1 Built-in self-test

During system startup, a built-in self-test (BIST) is carried out for each board.

If no defective boards are found during this test, the result is output as follows on the console:
SYSTEM CONFIGURATION AND BIST STATUS.....PASSED

If defective boards or board components are found during this test, the result is output on the console (e. g.),

RM600 E:

```
SYSTEM CONFIGURATION AND BIST STATUS.....FAIL□
.□
.□
.□
□□□□□□□□□□□□Bist Failure List□
Cab□□□Bus□□□Slot□□□Agent_no□□□Product□□□□□□Vendor□□□□□Test□□□□□Status□
-------------------------------------------------------------------------□
00□□□□00□□□□□02□□□□□□□□00□□□□□□□CP4-V02□□□□□□□□147□□□□□□□□038□□□□□Warning
```

RM600-xxx:

```
SYSTEM CONFIGURATION AND BIST STATUS.....FAIL□<===□
.□
.□
.□
□□□□□□□□□□□□Board Failure List□
Cabinet□□Slot□□□□Product□□□□Vendor□□□□Test□□□□Status□
□0□□□□□□□□10□□□□MMM-167xxx□□□147□□□□□□□□238□□□□□Failed□
□0□□□□□□□□13□□□□CPU-150402□□□147□□□□□□□□238□□□□□Failed
```

and the system responds as follows:

- The defective boards or components are deactivated, and system startup continues providing them do not prevent this.

  After the logging daemon starts, malfunctions are logged in Logging□V3.0 (...). The defective boards or components and their deactivation are entered in the HW_CONFIG data base (...) and are displayed in VConfig (...).

  In addition, these boards or components are flagged in the NVRAM as *defect*.

  NVRAM stands for non-volatile random access memory. It is located in the EHIOS or HIOS and, among other things, stores error information about boards so that any defective boards are not activated the next time the system is started up.

  ▶ Notify Service so that the defective boards or components can be replaced in due time.

- If all the boards installed on the SP bus are of the same type, or the defect is
  - in the HIOS board in slot 8 or the EHIOS board in slot 10 (RM600-E30) or 14 (RM600-E70) of the system cabinet of an RM600 E, or
  - in the CSI board of an RM600-xxx,

  then the system startup will be aborted at this point.

  ▶ Turn the system off and notify Service so that the defective boards can be replaced without delay.

CP4 boards (RM600 E) or memory boards (RM600-xxx) are activated as soon as the BIST has been completed because the memory modules on them must be available in order to load Reliant UNIX. This is why these boards which were flagged in the NVRAM as *defect* are given the *FAIL* status in the BIST so that they are not activated at all.

The only exception is in the case of the last CP4 or memory board that is available. So that the system can be powered up in an emergency or for test purposes (see the following section), this board is given the *PASS* status in the BIST although it was flagged as *defect* in the NVRAM.

2.1.3.2 Analyzing the NVRAM

Once the BIST has been performed, the information stored in the NVRAM is evaluated. The system checks which boards have passed the BIST without error but are flagged in the NVRAM as *defect*. First of all, it establishes the following:

- on an RM600 E, does the system have at least one functioning CPU as boot CPU, plus one CP4 board with a memory module, and the HIOS board in slot 8 or the EHIOS board in slot 10 (RM600-E30) or 14 (RM600-E70) of the system cabinet.
- on an RM600-xxx, does the system have at least one functioning CPU as boot CPU, plus either one MMM board as memory or one CPU board with on-board-memory, one MBP board as a bus bridge to the CSI board (or one channel if the MBP boards are double) and the CSI board itself.

- If this is the case, system startup continues.

  The system then checks whether there are any defective boards in addition to those mentioned above which are not to be activated.

  Any defective boards that are found on a RM600 E are deactivated and the following message is displayed on the console, e. g.:

```
□□□□□□□□□□□□LAR Failure List□
Cab□□□Cage□□□Slot□□□Chan1□□□□□□Chan2□□□□Serialnumber□
-----------------------------------------------------□
00□□□□□01□□□□□09□□□00000000□□□□0000□□□□□000000001083
```

  Any defective boards that are found on a RM600-xxx are deactivated and the following message is displayed on the console, e. g.:

```
CMN: WARNING: MBII 3 mb2_pre_init(): Skipping broken CPU board.□
MBII:55:□□□□□□ctlr_name :bus_type:cab_id:slot_id:test_id:test_stat:test_level□
MBII:1:□□□□□□□CPU-150401:SP□□□□□□:0□□□□□:12□□□□□:ee□□□□□:80□□□□□□□:4c
```

At the end of system startup, the defects table of the HW_CONFIG data base is output on the console. This includes a list of all defective and deactivated boards (...). The defective boards and their deactivation are also displayed in VConfig (...).

- Notify Service so that the defective boards can be replaced in due time.

- If this is not the case,
  - system startup on a RM600 E is stopped before boot1 and a message appears on the console (e. g.):
    *** WARNING: HIOS is LAR defect□
    □
    □□□□□□□□□□□□□□□□□Type in '1' ignore errors and boot system□
    □□□□□□□□□□□□□□□□□□□□□□□□□□'2' activate BDM□
    □□□□□□□□□□□□□□□□□□□□□□□□□□'3' power off system□
    □
    Your choice:
    - Enter **3** in response to this message. Notify Service so that the defective boards can be replaced without delay. The option of answering **1** or **2** is intended for Service engineers so that the system can be powered up with defective boards in an emergency or for test purposes.

      If the entire EHIOS or HIOS board is not defective but only the subcard to which the boot disk is connected, system startup does not stop until the beginning of boot2. A message is output on the console, e. g.:
      SLOT_ID: 8□□CHAN_ID: 1□
      Bootchannel is lar-defect□
      Do you want to continue y/n
    - Enter **n** in response to this message and turn the system off. Notify Service so that the defective board can be replaced without delay. The option of answering **y** is also intended for Service engineers.
  - system startup on a RM600-xxx is stopped before boot1 and the following message appears on the console:
    <*board*> is LAR-defect. Do you want to continue (Y/N)?
    - Enter **N** in response to this message and turn the system off. Notify Service so that the defective boards can be replaced without delay. The option of answering **Y** is intended for Service engineers so that the system can be powered up with defective boards in an emergency or for test purposes.

2.1.4 The boot1 phase

The boot1 phase is initiated when the board has been installed and started. It then loads boot2 in the main memory and starts it. Various errors can occur during the boot1 phase (...).

- In the case of minor errors, which do not prevent the system startup, an error message is output on the console and the system startup is continued.
- In the case of serious errors, which prevent the system startup, an error message is likewise output on the console and the system startup is aborted.

  On a RM600 E the following instruction is issued:

  1. Press Ctrl-@ BDM to return to BDM□
  2. Check Bootparameters□
  3. Enter "creset" command to restart the system

  On a RM600-xxx the following instruction is issued:

  1. Press Ctrl-A to return to BDM□
  2. Check Bootparameters□
  3. Enter "creset" command to restart the system

It can happen that a serious error is reported as a result of one or more previous minor errors. In this case, all messages must be considered in context in order to interpret the problem, e. g.

[ERROR] SD[<*scsi_bus*>,<*scsi_id**8>]: start unit failed□
[ERROR] Cannot open file <*file_name*>

Meaning:

The SCSI command *start unit* was not executed correctly so the file specified under *file_name* cannot be opened. System startup is terminated.

- If system startup is terminated because of a serious error, first check the boot parameters (see the "Reliant UNIX Operation" manual). Put them right if they incorrect and start the system again.
- If you cannot detect any errors in the boot parameters, or if system startup is terminated in the boot1 phase again, make a note of the error messages, turn the system off and notify Service.

2.1.5 The boot2 phase

The boot2 phase checks the NVRAM for inconsistencies and loads the Reliant UNIX system kernel into the main memory.

2.1.5.1 Failure of the boot disk

Boot2 cannot be loaded if the boot disk fails during system startup.

You see a message like the one below in the event of a boot disk failure (example for RM600 E):

Scanning disks using Disk-ID "SEAGATE ST31200N□□□□□□□□□00217678"□
..... □
[ERROR] No disk with DISK-ID "SEAGATE ST31200N□□□□□□□□00217678"□
1. Press Ctrl-@ BDM to return to BDM□
2. Check Bootparameters□
3. Enter "creset" command to restart the system

If your system is equipped with mirror boot disks, or if another system disk is installed in the system cabinet, then you can power up the system by switching to the mirror of the active boot disk or to the other system disk (...).

If your system does not have mirror boot disks and no other system disk is available, you can only load the latest system backup or reinstall the system (see the "Reliant UNIX Installation" manual).

2.1.5.2 Checking the NVRAM for inconsistencies

During the Boot2 sequence, the NVRAM is subjected to the following tests, one after another:

- Check whether the NVRAM contains correct data.

  Possible causes of inconsistencies:

  - The contents of the NVRAM have been corrupted (e. g. through loss of voltage).
  - The contents of the NVRAM have been initialized with the BDM command *nv_set*.

- The (E)HIOS board (RM600 E) or the CSI board (RM600-xxx) has been replaced but the NVRAM has not been initialized.

- Check whether the checksum of the NVRAM is consistent with the checksums of the NVRAM backups on the system disk.

  Possible causes of inconsistencies:
  - The NVRAM contains inconsistent data because the (E)HIOS board (RM600 E) or the CSI board (RM600-xxx) has been replaced (transfer of the NVRAM from another system).
  - No NVRAM backup was made after the last modification of the NVRAM (e. g. because it was impossible to shut down the system properly owing to a system halt).

- Check whether the checksum of the NVRAM is consistent with the checksum of the NVRAM backup on the boot disk.

  Possible causes of inconsistencies:
  - One of the first two checks reported an error but no action has been taken to put the situation right.
  - The root disk has been replaced, or the mirror of the active boot disk or another system disk has been accessed because the boot disk has failed.

In each of these cases you will see a warning on the console and the system start will be interrupted in Boot2.

You can read how to resolve all these inconsistencies in the chapter "Routines", section Resolving inconsistencies in the NVRAM (RM600).

2.1.5.3 Loading Reliant UNIX

The following problems may arise when loading Reliant UNIX.

- Loading hangs without an error message.

  The Reliant UNIX kernel has been corrupted.
  - Switch on the system again and load another system kernel (e. g. *unix.old*; ...).

- The load process is aborted with the following error message, for example:
  unix not found□
  ...□
  couldn't load dkncr(1,24,0)unix□
  boot2:

  The *unix* file cannot be found in the root partition of the boot disk because the root file system was not terminated correctly the last time it ran or because the *unix* file was deleted.

  - Load the Mini System (...). Go to a shell from the main menu of the Mini System and check the root file system (e. g.):

    fsck -F ufs -l /dev/ios0/sdisk000s0

    If the root file system cannot be restored, or if data is lost in the process, you should contact Service.

    If the root file system does not contain errors and has been restored, exit the Mini System via the following menu item:

    Abort Mini System and Reboot

  - If loading is aborted again with the same message, load another system kernel (e. g. *unix.old*).

2.1.6 Starting Reliant UNIX

When the system kernel has been loaded by boot2, control passes to Reliant UNIX and the system is powered up in the operating state that is defined in the */etc/inittab*(4) file.

The messages you receive during the startup process depend on the hardware and software configuration of your system. Any Reliant UNIX errors that occur in this phase are handled in exactly the same way as when the system is up and running (...).

During the startup, the root file system is mounted, defective SCSI devices are deactivated, the system is initialized with the init process and the firmware and hardware revision levels are checked for consistency. Problems and messages that may arise during these startup phases are explained in greater detail below.

2.1.6.1 Mounting the root file system

The following problem may arise when mounting the root file system:

- The startup is aborted with the following error message:
  cannot mount root filesystem

  The root file system cannot be mounted because it is damaged.
  - Load the Mini System (...). Go to a shell from the main menu of the Mini System and check the root file system (e. g.):

    fsck -F ufs -l /dev/ios0/sdisk000s0

    If the root file system cannot be restored, or if data is lost in the process, you should contact Service.

    If the root file system does not contain errors and has been restored, exit the Mini System via the following menu item:

    Abort Mini System and Reboot

2.1.6.2 Deactivating SCSI devices

If any defects occur in SCSI devices while the system is running and these can result in loss of data,

- then the devices are flagged as *defect* in the file */etc/opt/lar/scsi_dev.defect*.
- then the malfunctions are logged in Logging V3.0 (...), and the defective SCSI devices are entered in the HW_CONFIG data base (...) and are displayed in VConfig (...).
- then, during the next system startup , at the latest, these devices are not activated again (...) and the following message is displayed on the console, e. g.:
  /sbin/scsilar_rc: deactivate ios0/sdisk072
- then, after the logging daemon has started, deactivation is logged in Logging V3.0, entered in the HW_CONFIG data base and displayed in VConfig.

2.1.6.3 Initializing the system with the init process

The main functions of the init process (Detailed information on initializing the system can be found in the "Reliant UNIX Operation" manual.) include creating further processes from the entries in the */etc/inittab*(4) file, starting procedures (e. g. *rc* procedures; see *rc*(2)) and starting login services for the console and the local connections.

Problems that may arise when the system is being initialized fall into three categories:

- Incorrect initialization steps without aborting system startup

  In such cases, you have to be prepared for individual components to malfunction when the system is up and running.

- Hangup situations which do not resolve themselves
- Incorrect initialization steps with system startup aborting

Problem analysis is mainly confined to evaluating "suspicious" messages and pinpointing the time of the error. The error messages displayed immediately prior to the error event should be given most attention. These messages should be jotted down if possible. This is especially true if system startup is aborted or a hangup situation occurs so there is no access to the system and its log files.

During the initialization phase it is often impossible to locate error messages precisely because the screen is scrolled quickly and many messages are output. Furthermore, error messages can often barely be distinguished from status information. In these circumstances it is advisable to evaluate the log by means of Logging□V3.0 (...), VConfig (...) or the diagnosis menu (...). Additional diagnostic resources are also available to assist you with the evaluation, where necessary (...).

The problems described can arise during the initialization phase. Notify Service if a suggested approach does not solve a problem.

Incorrect initialization steps without aborting system startup

- Errors are reported from *rc* procedures.

  The hardware or software has been configured incorrectly, or not at all.
  - Make a note of the error messages and evaluate the log files. Find the errors and correct them.

  The hardware is malfunctioning.
  - Check whether all devices have been switched on and connected correctly.

- The following error message is displayed on the console:
  INIT: Command is respawning too rapidly. Check for possible Errors.
  id: *</etc/inittab_entry>*

  The entry referred to in the */etc/inittab* file has been executed incorrectly on more than one occasion (e. g. because of a malfunctioning input/output interface).
  - Check the entry in the file and eliminate the cause of the error message. Then check the entire file using the *init q* command.

- The following error message is displayed on the console:
  <*rc_procedure*> failed with exit status of 100.
  SYSTEM is going to SINGLE USER.
  Check the script and restart.

  A serious error has occurred in the specified *rc* procedure. The system is powered up in single-user mode, and you must enter the *root* password to get a shell:
  Type Ctrl-d to proceed with normal startup,
  (or give root password for system maintenance):
  - Check the bad *rc* procedure, possibly together with Service, eliminate the cause of the error and power up your system again with *init 6*.

    To be able to work with the system on a restricted basis at least, you can also try to power up the system in multi-user mode without this procedure. To do this, rename the file and power up your system again:

    mv /etc/rc2.d/S69inet /etc/rc2.d/s69inet
    init 6

Hangup situations which do not resolve themselves

- System startup hangs during the initialization phase without an error message.

  There is a hardware malfunction, or the software is inconsistent, buggy or configured incorrectly, or an individual process that was started from an *rc* procedure is hanging.
  - Take note of the preceding messages and attempt to reboot your system.

- System startup is halted with the following message at the start of the init process:
  ENTER RUN LEVEL(0-6,s or S):

  The *initdefault* entry is missing from the */etc/inittab* file.
  - Specify a valid operating state (generally **2** for multi-user mode) and, when the system has powered up, insert the *initdefault* entry at the following position in the file, e. g.:

    brc:1234:bootwait:/sbin/brc >/dev/console m2>&1 </dev/console
    is:2:initdefault:
    rS:S:wait:/sbin/rcS >/dev/console 2>&1 </dev/console

Incorrect initialization steps with system startup aborting

- The system is automatically rebooted when the root file system is checked:
  The root file system (<*file_system*>) is being checked
  ...SYSTEM WILL REBOOT AUTOMATICALLY

  The root file system has been corrupted or is inconsistent.
  - The system repairs itself. You do not need to intervene in any way.

- Should the problem recur despite repeated efforts to resolve it, load the Mini System (...). Go to a shell from the main menu of the Mini System and check the root file system (e. g.):

  fsck -F ufs -l /dev/ios0/sdisk000s0

  If the root file system cannot be restored, or if data is lost in the process, you should contact Service.

  If the root file system does not contain errors and has been restored, exit the Mini System via the following menu item:

  Abort Mini System and Reboot

- System startup is aborted directly after the start of the init process, and there is an immediate restart.

  The operating state has not been specified in the *initdefault* entry in the */etc/inittab* file.

  - Power the system up in single-user mode (...), and enter a valid operating state (generally **2** for multi-user mode):

    is:**2**:initdefault:

- System startup is aborted with one of the following messages:
  /usr sub-tree is not present□
  /var sub-tree is not present

  The */usr* or */var* file system is inconsistent and has therefore not been mounted.

  - Load the Mini System (...). Go to a shell from the main menu of the Mini System and check the inconsistent file system. The file system can be of type *vxfs* (following a reinstallation) or *ufs* (following an update installation), e. g.: (Different information may need to be entered for the partition in your system. The assignments are defined in the */etc/vfstab* file.)

    fsck -F vxfs -l /dev/ios0/sdisk000s3□□□
    (for /usr)□
    fsck -F vxfs -l /dev/ios0/sdisk000s4□□□
    (for /var)□
    □
    fsck -F ufs -l /dev/ios0/sdisk000s3□□□□
    (for /usr)□
    fsck -F ufs -l /dev/ios0/sdisk000s4□□□□
    (for /var)

     If you specified the incorrect type, the system responds with invalid super block magic.

    If the file system cannot be restored, or if data is lost in the process, you should restore the last system backup.

    If the file system does not contain errors and has been restored, exit the Mini System via the following menu item:

    Abort Mini System and Reboot

- Messages are displayed during the system start regarding manually checking a file system, e. g.:
  /dev/ios0/rsdisk000s5: 0 BAD I=12547□
  /dev/ios0/rsdisk000s5: UNEXPECTED INCONSISTENCY, RUN fsck MANAULLY.□
  ufs mount: /dev/ios0/sdisk000s5: CONSISTENCY ERROR (This message contains the ufs or vxfs entry depending on the file system type.)□
  ufs mount: /dev/ios0/sdisk000s5 (Filesystem is dirty [run fsck]).□
  ...□
  Please check /home file system

  The reported file system is inconsistent or damaged. The system will be booted in single-user mode.
  - Enter the root password to get a shell:
    Type Ctrl-d to proceed with normal startup,□
    (or give root password for system maintenance):

    Check the file systems. You can use the *fsck*(1M) command to check all file systems contained in the */etc/vfstab* file. Clean, mounted file systems are skipped.

    The repaired file systems are still marked "dirty" after the first run through. You should consequently invoke the *fsck* command a second time.

    You can then boot the system in multi-user mode, using the *init 2* command (see *init*(1M)).

- System startup is aborted with the following message:
  S03swap: /etc/vfstab not found

  The */etc/vfstab* file does not exist.
  - Restore the file from the last system backup.

2.1.6.4 Checking consistency of firmware and hardware revision levels

The consistency check involves using the *fw_check*(1M) command to check the firmware and hardware revision levels of the controllers as well as the firmware revision levels of the EIPs in the individual cabinets.

Inconsistent firmware and hardware revision levels can impair the functioning of the system. Messages relating to inconsistencies indicate that the boards or EIPs in question need to be replaced or that the EEPROM firmware needs to be updated (see below).

Checking the controllers

The firmware revision levels of the controllers of one type are compared with each other, and the firmware and hardware revision levels of the installed controllers are compared with entries in the reference file named *fw_ref*. In addition, the system checks whether the firmware revision levels are numeric.

The reference file is generated automatically during the installation procedure, and is updated, if necessary, in the course of update installations.

What happens with inconsistencies that are detected:

- They are reported on the console, e. g.:
  fw_check 13:cab_id:2:slot:8:bus:MBII: hardware rev. for module SIH-195700 is 2, should be 3, also firmware rev. 01.20, should be 01.22
- They are reported to Logging□V3.0 (...) when the logging daemon has started and are entered in the HW_CONFIG data base (...) with the *error* attribute. The inconsistencies are also displayed in VConfig (...).
- Notify Service in the event of inconsistent hardware revision levels so that the boards can be replaced without delay.
- As far as inconsistent firmware revision levels are concerned, update the EEPROM firmware (see section "Preparing for installation - Updating the EEPROM firmware" in the "Reliant UNIX Installation" manual).

Checking the EIPs

The system checks whether the EIPs in the individual cabinets have the same firmware revision level, and whether the firmware revision levels are numeric.

What happens with inconsistencies that are detected:

- They are reported on the console, e. g.:
  fw_check 20:cab_id:0:hw_rev:2:cab_type:SYSTEM_HE:serial:SY56100201: firmware rev. for EIP with hardware rev. 2 is not numeric: 3.0.fa
- They are reported to Logging□V3.0 (...) when the logging daemon has started.

▶ If firmware revision levels are inconsistent, notify Service so that the EIPs can be replaced without delay.

List of error situations

There are the following error situations:

- The firmware revision level of a controller is not numeric. It may be a development level (*fw_check 1*):
  firmware rev. for module <*controller*> is not numeric: <*fw_stand*>
- The firmware revision levels for controllers with the same module numbers and version numbers are different. The highest firmware revision level is assumed to be correct and all lower levels are reported (*fw_check 2* and *fw_check 3*, the latter for CPUs):
  firmware rev. for module <*controller*> is <*fw_stand*>, should be <*fw_stand*>
- The firmware revision levels for controllers with the same module numbers but different version numbers are different (*fw_check 4*):
  firmware rev. are different, module <*controller*> has <*fw_stand*>, module <*controller*> has <*fw_stand*>. Keep firmware revisions equal
- The firmware revision level of a controller is less than the reference specification in the *fw_ref* file (*fw_check 11*):
  firmware rev. for module <*controller*> is <*fw_stand*>, should be <*fw_stand*>
- The hardware revision level of a controller is less than the reference specification in the *fw_ref* file (*fw_check 12*):
  hardware rev. for module <*controller*> is <*hw_stand*>, should be <*hw_stand*>
- The hardware and firmware revision levels of a controller are less than the reference specifications in the *fw_ref* file (*fw_check 13*):
  hardware rev. for module <*controller*> is <*hw_stand*>, should be <*hw_stand*>, also firmware rev. <*fw_stand*>, should be <*fw_stand*>
- The hardware and firmware revision levels of a controller are less than the reference specifications in the *fw_ref* file and the firmware revision level is less than the highest level (*fw_check 14*):
  hardware rev. for module <*controller*> is <*hw_stand*>, should be <*hw_stand*>, also firmware rev. <*fw_stand*>, should be <*fw_stand*>, max. value of firmware ref. is <*fw_stand*>
- The firmware revision level of a controller is less than the reference specification in the *fw_ref* file and less than the highest level (*fw_check 15*):
  firmware rev. for module <*controller*> is <*fw_stand*>, should be <*fw_stand*>, max. value of firmware ref. is <*fw_stand*>
- The firmware revision level of an EIP is not numeric. It may be a development level (*fw_check 20*):
  firmware rev. for EIP with hardware rev. <*hw_stand*> is not numeric: <*fw_stand*>
- The firmware revision levels of EIPs with the same hardware revision level are different. The highest firmware revision level is assumed to be correct and all lower levels are reported (*fw_check 21*):
  firmware rev. for EIP with hardware rev. <*hw_stand*> is <*fw_stand*>, should be <*fw_stand*>

2.1.7 Outputting the HW_CONFIG defects table

The defects table of the HW_CONFIG data base is output on the console immediately before the message The system is ready. It contains information on errored, defective and deactivated hardware components as well as inconsistent firmware and hardware revision levels of controllers.

This information is also output on the screen whenever anybody logs on as *root* (at the login or using *su - root*) and, in a more differentiated form, when you issue the command *show_lar*(1M).

HW_CONFIG defects table:□
□
type□□:name□□□□□□□□:log:cab:bid:slt:chn:attribute:set□by□□□□□□:cause□
Fan□□□:(hw address□)□□□:0□□:□□□:1□□:□□□:defect□□□:system□□□□□□:ubft□3042□
SPbus□:CP4-V02□□□□□:□□□:0□□:0□□:2□□:2□□:deactive□:set_defects□:deactivated by LA□

SPbus□:CP4-V02□□□□□:□□□:0□□:0□□:2□□:2□□:defect□□□:set_defects□:set by set_defect□
SPbus□:HIOS□□□□□□□□□:□□□:0□□:0□□:9□□:□□□:deactive□:set_defects□:deactivated by LA□
SPbus□:HIOS□□□□□□□□□:□□□:0□□:0□□:9□□:□□□:defect□□□:operator□□□□:set by operator□
SPbus□:HIOS□□□□□□□□□:□□□:0□□:0□□:9□□:0□□:deactive□:set_defects□:deactivated by LA□
SPbus□:HIOS□□□□□□□□□:□□□:0□□:0□□:9□□:0□□:defect□□□:set_defects□:set by set_defect□
SPbus□:CP4-V02□□□□□:□□□:0□□:0□□:0□□:0□□:error□□□□:fw_check□□□□:fw_check 11□
SPbus□:CP4-V02□□□□□:□□□:0□□:0□□:2□□:0□□:error□□□□:fw_check□□□□:fw_check 11□
Disk□□:ios0/sdisk072(MP47)□□□□□□□□□□□□□□□:deactive□:system□□□□□□:set□by□LAR□
Disk□□:ios0/sdisk072(MP47)□□□□□□□□□□□□□□□:defect□□□:system□□□□□□:EE_DISK 2□
Disk□□:ios0/sdisk101(MP25)□□□□□□□□□□□□□□□:error□□□□:system□□□□□□:EE_DISK 1

The individual columns in this output have the following meaning:

type This column shows the type of the component (e. g. FAN, SPbus or Disk).

name
: This column shows the component (e. g. the module number for controllers, the name of the device file or the device type for SCSI devices).

log This column shows the logical controller number.

cab, bid, slt, chn
: These columns show the location of controllers (cabinet, bus ID, slot and channel).

attribute
: This column shows the status of the component:

  error The component is errored but remains in operation.

  defect
  : The component is defective and must be replaced. It will be deactivated - at the latest - the next time the system is booted.

  deactive
  : The component has been deactivated.

set by
: This column shows who or what made the entry (operator, system or software component).

cause
: This column shows the cause of an entry. If it indicates an error (e. g. *fw_check 11* or *EE_DISK 2*), you can use the *getemm*(1M) command to see a description of the problem and handling instructions to solve it.

2.1.8 Login

The problems described below may arise when you log in to the system. Notify Service if a suggested approach does not solve a problem.

- The login prompt appears again once the first character of the user ID has been input:

  The *autopush* entry is missing in the */etc/inittab* file. Different entries are required, depending on the type of connection:

  ap::sysinit:/sbin/autopush -f /etc/ap/cons.ap□□□□□□□□# Console□
  si::sysinit:/sbin/autopush -f /etc/ap/si.ap□□□□□□□□□□# SIM controller□
  sih::sysinit:/sbin/autopush -f /etc/ap/sih.ap□□□□□□□□# SIH controller□
  dptg::sysinit:/sbin/autopush -f /etc/ap/dptg2.ap□□□□□# DPTG2 protocol□
  rtty::sysinit:/sbin/autopush -f /etc/ap/rtymux.ap□□□□# Remote TTY

  - Provided the entry does not relate to the console, log in as *root* and correct the */etc/inittab* file.
  - If the console entry is missing, load the Mini System (...). Go to a shell from the main menu of the Mini System and mount the root file system (e. g.):

    mount -F ufs /dev/ios0/sdisk000s0 /svr4

    Correct the */svr4/etc/inittab* file.

    Exit the Mini System via the following menu item:

    Abort Mini System and Reboot

  The required Streams modules have not been configured.

  - Provided the entry does not relate to the console, log in as *root* and invoke the following command:

    autopush -f /etc/ap/*.ap (The **"*"** in the command line stands for the connection type.)

  - If it is the entry for the console, restore the file from the last system backup.

- The login prompt appears again when the user ID has been input.

  The */dev/tty* device file has been set up with the wrong major and minor numbers, or the device file does not exist.

  - Load the Mini System (...). Go to a shell from the main menu of the Mini System and mount the root file system (e. g.):

    mount -F ufs /dev/ios0/sdisk000s0 /svr4

    Delete the */svr4/dev/tty* file if necessary and recreate it:

    mknod /svr4/dev/tty c 130 0

    Exit the Mini System via the following menu item:

    Abort Mini System and Reboot

  The */etc/ap/*.ap* file is empty.

  - Restore the file from the last system backup.

  The access permissions of the root file system have been changed with the result that the user no longer has read and execute rights.

  - Log in as *root* and check the access permissions:

    ls -ld /

    The access permissions should be set as follows:
    drwxr-xr-x 40 root root 1536 Apr 27 05:19 /

    Correct the access permissions if necessary:

    chmod 755 /

- No prompt appears when the password has been input.

  If other users can still log in, the error probably lies in the *$HOME/.profile* file.

  - Log in as *root*, and correct the *.profile* file of the user in question. If you cannot log in as *root*, you can bypass the *.profile* of the *root* user by logging in under another user name and issuing the *su root* command to get root permissions.

  If no user can log in, the error probably lies in the */etc/profile* file.

  - Load the Mini System (...). Go to a shell from the main menu of the Mini System and mount the root file system (e. g.):

    mount -F ufs /dev/ios0/sdisk000s0 /svr4

    Correct the */svr4/etc/profile* file.

    Exit the Mini System via the following menu item:

    Abort Mini System and Reboot

- The login prompt and screen inputs are garbled or contain non-readable characters.

  The */etc/ap/cons.ap* file does not exist or has been corrupted.

  - Restore the file from the last system backup.

- Characters that have been keyed in do not appear on the screen.

  The screen output has been halted.

  - Enable screen output again by pressing [CTRL][Q].

- The following message is displayed and the login prompt appears again:
  Unable to change directory to "<*directory*>"

  The directory assigned to the user ID in the */etc/passwd* file does not exist or its access permissions are incorrect.
  - Log in as *root*, check the entry in the */etc/passwd* file, and the access permissions for the directory, and correct them as appropriate (see *chmod*(1), *chown*(1), *chgrp*(1)).

  The directory assigned to the user ID in the */etc/passwd* file is not mounted.
  - Log in as *root* and execute the *fsck*(1M) and *mountall*(1M) commands (both without options).

  If you cannot log in as *root*, you can bypass the *.profile* of the *root* user by logging in under another user name and issuing the *su root* command to get root permissions.
- The following message is displayed and the login prompt appears again.
  No Shell

  The executable program assigned to the user ID in the */etc/passwd* file cannot be loaded.
  - Log in as *root*, check the owner and access permissions of the program and the associated path components, and correct them as appropriate.

  The directory assigned to the user ID in the */etc/passwd* file is not mounted.
  - Log in as *root* and execute the *fsck* and *mountall* commands (both without options).

  If you cannot log in as *root*, you can bypass the *.profile* of the *root* user by logging in under another user name and issuing the *su root* command to get root permissions.

- The following message is displayed when the user ID and password have been input:
  Login incorrect

  An incorrect user ID and/or password has/have been input.
  - Enter a valid user ID and password at the next login prompt.

    The actual keyboard allocation may not match the labeling on the keyboard because of the way your keyboard has been configured. In this case, some characters (special characters in particular) will have been moved to other keys.

    If you cannot remember the password, log in as *root* and delete the password:

    passwd -d -f <*user_id*>

    At the next login, the user is prompted to specify a new password.
  - If you cannot remember the password of the *root* user load the Mini System (...). Go to a shell from the main menu of the Mini System and mount the root file system (e. g.):

    mount -F ufs /dev/ios0/sdisk000s0 /svr4

    Delete the password from the */svr4/etc/shadow* file.

    Exit the Mini System via the following menu item:

    Abort Mini System and Reboot

  The */var/adm/lastlog* file is inconsistent.
  - Reboot the system.

- The login attempt is denied with the following message:
  rlogind: cannot create utmp entry

  There is no more free space in the */var* file system so no more log entries can be written to the */var/adm/utmpx* file.
  - If you have logged in as *root*, clean the */var* file system by deleting superfluous files (e. g. in */var/tmp*). You may have to set up a larger partition for */var* (see "System Administrator's Guide").
  - If you are not logged in as *root* and no other users are logged in so you cannot get root permissions with *su root*, you have to load the Mini System (...). Go to a shell from the main menu of the Mini System and mount the root file system (e. g.):

    mount -F ufs /dev/ios0/sdisk000s0 /svr4

    Clean the */svr4/var* file system.

    Exit the Mini System via the following menu item:

    Abort Mini System and Reboot

- The login process takes an unusually long time.

  The */var* file system is possibly very full.
  - Use the *df -k* command (see *df*(1)) to check the utilization of the */var* file system.
  - If the file system is more or less full, clean it as described under the item above.

2.2 System running

2.2.1 System environment monitoring

While the system is running, the system environment is monitored by EIPs which keep track of the computer, the ambient temperature, the functioning of fans and the BBU. With the exception of the UPS cabinet each cabinet has its own EIP. EIPs in the expansion cabinets (RM600 E), MBII and peripheral cabinets (RM600-xxx) and BBU cabinets report malfunctions to the EIP in the system cabinet which is responsible for passing on malfunction reports and reacting to malfunctions.

Any malfunctions of the above-mentioned components are reported as follows:

- On an RM600□E, the EIP reports the malfunction to the HIOS controller in slot 8 or the EHIOS controller in slot 10 (RM600-E30) or 14 (RM600-E70) of the system cabinet. Malfunctions of the UPS are detected by the EIP in the cabinet to which the USP is connected via a contact interface.
- On an RM600-xxx, the EIP reports such malfunctions to the CSI controller. Malfunctions of the UPS are detected by the CSI controller directly.

The (E)HIOS or CSI controller reports malfunctions to the UBFT driver which in turn notifies the UBFT daemon (see also *ubft*(7) and *ups*(1M)).

What happens with malfunctions:

- They are reported on the console, e. g.:
  DRV: Err1: ubft 3042 (48,0) (daemon) cab 5/17: back-left fan ventilation unit single failure!

  You can use the *getemm*(1M) command to see a description of the error and handling instructions to eliminate it.
- They are handled by the UBFT daemon as defined by the parameters in the *_upstab*(4) file which you can set to meet the requirements of your system.

- They are written to the */var/adm/ubft/log* file.

  In critical situations, if the system has not shut down within a period of time specified by the EIP, then after a few minutes the EIP will automatically switch the system or the offending cabinet off. Since this is not an orderly shut-down it may result in loss of data. In this case you should power the system down. The latest reason for the EIP to turn off a cabinet or the system can be seen in the file */var/adm/ubft/log*, provided the UBFT daemon is running when you start the system up again.

- They are reported to Logging□V3.0 (...), entered in the HW_CONFIG data base (...), and displayed in VConfig (...).
- They are indicated by the LCD and/or LED on the cabinet in question.

  RM600 E:

  - The *ON/BATT* LED on the system cabinet or on an expansion cabinet is red (with BBU only).

    The BBU of the cabinet is active due to a power failure or power pack fault.
  - The *ON/BATT* LED on a BBU cabinet is red.

    Power failure or fault in the power pack of the BBU cabinet.
  - The LCD on a cabinet indicates TEMP FAIL.

    The room temperature is too high or the cabinet's ventilation ducts are obstructed.
  - The LCD on a cabinet indicates FAN FAIL.

    There has been a total failure of the fans.

RM600-xxx:

- The *ON/BATT* LED on the system cabinet or the *ON/BATT* and *FAIL* LEDs on an expansion or peripheral cabinet is/are red (only in cabinets of the 2xx, 3xx and 4xx models with an internal BBU).

  The internal BBU of the cabinet is active due to a power failure or power pack fault.
- The *ON/BATT* and *FAIL* LEDs on the BBU cabinet are red.

  The BBU is active due to a power failure or a fault in a power pack of the system cabinet, an expansion cabinet or a peripheral cabinet.
- The LCD on the system cabinet indicates TEMP FAIL and/or the *FAIL* LED on another cabinet is red.

  The room temperature is too high or the cabinet's ventilation ducts are obstructed.
- The LCD on the system cabinet indicates FAN FAIL and/or the *FAIL* LED on another cabinet is red.

  There has been a total failure of the fans.

▶ Remedy the malfunction or notify Service so that defective components can be replaced.

You can find further details on system environment monitoring and the functioning of the UBFT daemon in the "Power failure or critical temperatures" chapter of the "Reliant UNIX Operation" manual.

2.2.2 Reliant UNIX error messages

2.2.2.1 Error handling

Reliant UNIX errors are handled by the system as follows.

- System errors (except of CPU board cache errors and memory board errors, see below), unrecoverable SCSI device errors and application errors which do not cause a system halt are written to the STREAMS device */dev/log* by error reporting routines (e. g. *cmn_err()*, *drv_err()* or *syslog*(3C)). The *syslogd*(1M) daemon reads */dev/log*.

  Applications which incorporate the Logging V3.0 application programming interface report errors directly to Logging V3.0 without using the syslog daemon (see section Passing data directly to Logging).

  The Syslog daemon handles these errors in accordance with parameters which are defined in the *syslog.conf*(4) file and which you can adjust to your requirements. Errors are reported on the console, written to various log files (e. g. */var/adm/log/messages*) and reported to Logging V3.0 (...), depending on the entries in the *syslog.conf* file.
- Recoverable SCSI device errors are written to the STREAMS device */dev/error* which is read by the *errord*(8) daemon. The error daemon stores these messages in the */var/adm/error/errfile.current.Z file.* Older files are given the name *errfile.previous.Z* and later *errfile.<date>.Z.* A user interface is at your disposal to evaluate these messages (...). In addition, the errors read by the error daemon are passed to the *errlogd*(8) daemon which monitors threshold limits for SCSI devices.
- Memory boards and CPU boards are monitored by the program *testecc*(1M). Single-bit and multi-bit errors which occur in memory boards, and single-bit error which occur in the caches of CPU boards, are passed to the error log daemon *errlogd*(8) which monitors the relevant thresholds.
- The error daemon distinguishes between two types of threshold limits:
  - Soft limit: The device is errored but remains in operation.
  - Hard limit: The device is defective and has to be replaced.

  When one of these threshold limits is reached, the error is reported to Logging V3.0 and an entry is made in the HW_CONFIG data base (...). A user interface is available to display manufacturers' specifications for SCSI devices threshold limits (...).
- All error messages reported to Logging V3.0 are collected in various log files and may be passed on to Event Management, DSM Event Reporting or TransView.

  In response to error messages, Event Management can start reaction programs which define the handling of the error message (...). For example, the *Set Controller Attribute "defect"* can be called to make an entry in the HW_CONFIG data base and the controller will not be started at the next boot.

  DSM Event Reporting is a separate program package for error diagnostics in SNA networks (...).

  TransView is a separate program package for system and network administration, and administration of applications in complex networks (...).

2.2.2.2 Structure of error messages

Reliant UNIX errors are output on the console with the following syntax:
[*reporting component*:] [*error weight*:] [*error identifier*] *error text*

Example:
DRV: Err1: ubft 3060 (48,0) (daemon) cab 5: non-redundant fan ventilation failure!

The reporting component is only displayed if the error is reported via the *cmn_err()* (CMN = kernel) or *drv_err()* (DRV = driver) error reporting routine.

The error weight is also only displayed if the error is reported via the *cmn_err()* or *drv_err()* error reporting routine.

Kernel	Driver	Display when	Meaning

		getemm is called	
PANIC	Fatal	EMERG	System halt
-	Err1	ALERT	Error indicating considerable loss of data and requiring immediate action
-	Err2	CRIT	Error indicating loss of data or malfunctions requiring action soon
-	Err3	ERR	Error indicating malfunctions but not requiring immediate action
WARNING	Err4	WARNING	Warning that normal functioning of the system could be impaired unless the appropriate action is taken
NOTICE	Warn	NOTICE	Indication of a possible problem
-	Note	INFO	Status message
-	Dbg1-Dbg5	DEBUG	Messages only used in program debugging

If an error identifier is output, you can use the *getemm*(1M) command to see a description of the problem and handling instructions to solve it. An error identifier always consists of the name of the reporting module in the component (ubft in the example above) and a unique error number relating to the module (3060 in the example above).

Example of the *getemm* call and its output:

getemm ubft:3060□
Text:□
non-redundant fan ventilation failure!□
Description:□
The Environment Interface Processor (EIP) indicates□
a non-redundant fan ventilation failure of the□
specified component.□
Action:□
Exchange or repair fan ventilation unit(s).□
Weight:ALERT

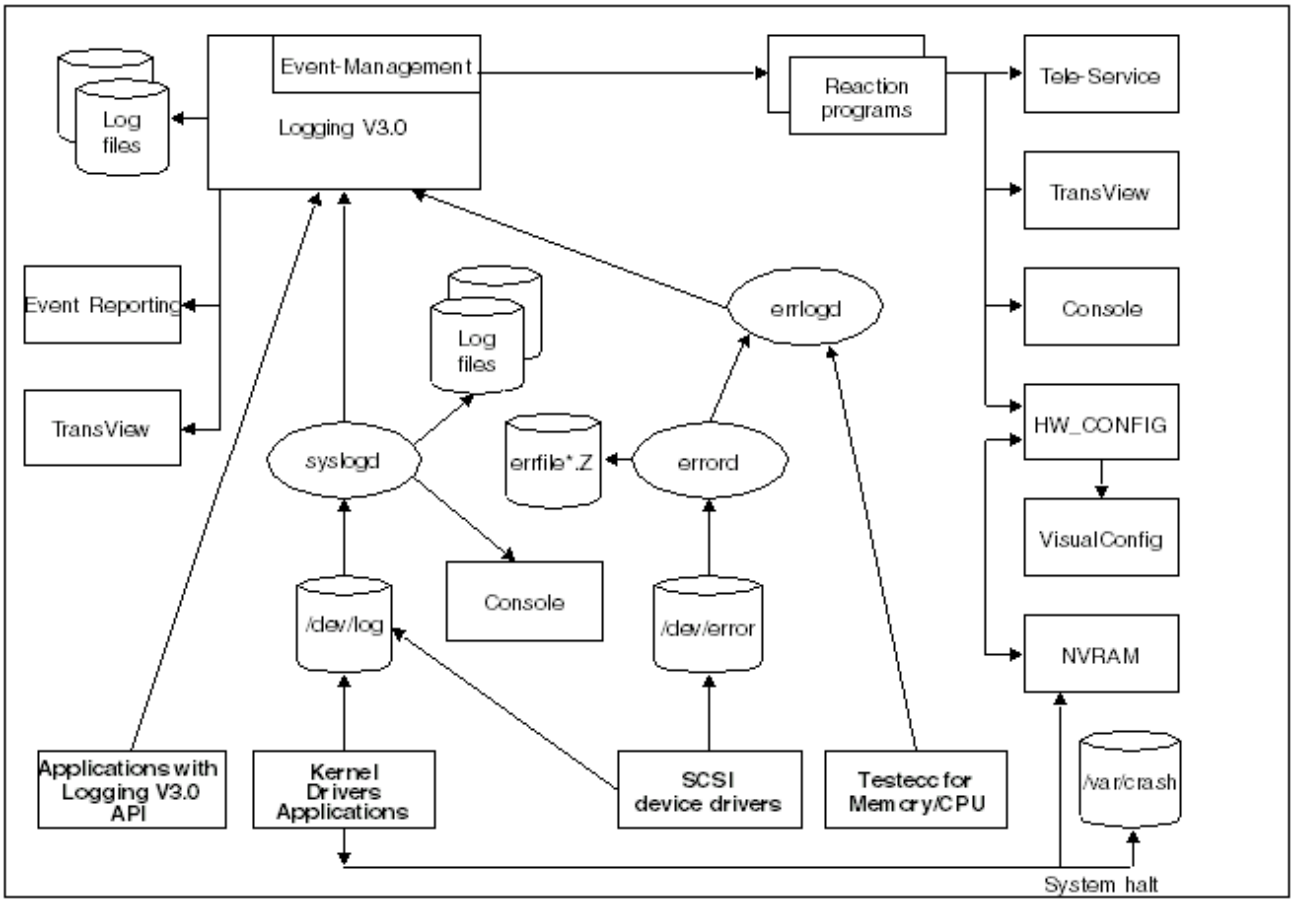


Figure 1: Data flow of error messages

2.2.3 System halts

If an error causes a system halt, this can be a serious software or hardware error, or a result of irregularities in operating system software that can be remedied by restarting the system. A system halt is primarily a protective function to avoid loss of data.

Here is the procedure that is followed in the event of a system halt.

- If the reason for the system halt is a defective board, the offending board is flagged as follows:
  - on an RM600 E, it is flagged *defect* in the NVRAM of the HIOS board in slot 8 of the system cabinet, or in the NVRAM of the EHIOS board in slot 10 (RM600-E30) or 14 (RM600-E70) of the system cabinet.
  - on an RM600-xxx it is flagged *defect* in the NVRAM of the CSI board.

  This prevents the defective board from being activated during the next system startup. In addition, the console log is saved in the NVRAM for logging purposes.
- The following message is displayed on the console, e. g.:
  CMN: PANIC: MBII 2 MBII: Bad firmware config data: couldn't find all CPU's
- The *ikdb*(8) debugger is started:
  You are now in the kernel debugger. Type ex to exit.□
  ikdb>
- Exit the debugger with the *ex* command. The system is then automatically shut down and powered up again.

What happens during shutdown:

- The system buffer is written back (see *sync*(1M)).
- A core dump is written to the swap area.

What happens during power-up:

- The core dump (*dump.<x>*), the file (*unix.<x>*) containing the executable system kernel, and an archive (*info.<x>*) (The "x" in the *dump*, *unix* and *info* file names stands for a serial number.), containing a series of system files in compressed form, are written to the standard directory */var/crash*. This directory is defined by the *DUMPDIR* variable in the */etc/default/dumpsave* file (see *dumpsave*(4)). You can modify it (see *dumpsave*(1M)).
- The console log that was saved in the NVRAM is read by the *syslogd*(1M) daemon and written to Logging□V3.0 (...).

On computers of type RM600 E that are equipped with EHIOS boards and more than 4 Gigabytes of main memory, a memory dump may be larger than the available space in the standard directory */var/crash*. For this reason, you should use the command *initdumpfs*(1M) on such computers to create a separate file system containing enough space for memory dumps, i.e. at least 1/6 of the total memory size.

If you have not yet created such a file system, then the Init process will output the following warning when the system is initialized:

DUMPPARTITION isn't set correctly in /etc/default/dumpsave□
On systems with EHIOS a special dumpdevice (extra mounted file system)□
must be configured! This will be done by executing once:□
□□□□□□□□/sbin/initdumpfs

Please save the *dump.<x>* and *unix.<x>* files and the *info.<x>* archive to an external medium (...), and forward this to the Service department. You should also include a short description of the circumstances in which the system halted.

If a kernel other than *unix* was active when the system halted (e. g. *unix.old*), then this kernel must be included in the error documentation because the core dump cannot be evaluated otherwise.

2.2.4 System hangups

If you find that your screen no longer responds to inputs without an error message being output on the console, there can be the following reasons for this.

- The system does not have any power or the connection between the screen and the computer or keyboard is disrupted.
  - Check the power supply of the system and the cabling of the screen.
- The screen is not set up correctly.
  - Check the local and system-based terminal configuration.
- The system is blocked because of an error or an application is hanging.
  - Try to log in at another screen as the *root* user. If you manage to do this, first save all data that is in memory to disk using the *sync*(1M) command. Then try to shut down the system properly.
  - If you cannot log in as the system administrator or if the system is blocked, try to start the *ikdb*(8) debugger on the console. To do this, press the [CTRL] key and enter "@". Release the [CTRL] key and directly enter (without any blank) the characters **IKDB** (in upper case). Terminate all application processes that are running and exit the debugger:

    ikdb> **ki**□
    ikdb> **ex**

    If the debugger cannot be exited, enter the *pa* command. If the system is still blocked after exiting the debugger, call the debugger again, enter the *pa* command and exit the debugger:

    ikdb> **pa**□
    ikdb> **ex**

    Exiting the debugger now triggers a system halt and control returns to the debugger. Exit the debugger again:

    ikdb> **ex**

    The following happens if one of the two actions (terminating all running processes or triggering a system halt) is successful:
    - The system buffer is written back (see *sync*(1M))
    - A core dump is written to the swap area
    - The system is automatically restarted. During the restart, the core dump (*dump.<x>*), the file (*unix.<x>*) containing the executable system kernel and a file (*info.<x>*) (The "x" in the *dump*, *unix* and *info* file names stands for a serial number.) containing all major system files are written to the standard directory */var/crash*. This directory is defined in the *DUMPDIR* variable in the file */etc/default/dumpsave* (see *dumpsave*(4)). You can modify it (see *dumpsave*(1M)).

      On computers of type RM600 E, which are equipped with EHIOS boards and more than 4 Gigabytes of main memory, a memory dump may be larger than the available space in the standard directory */var/crash*. For this reason, you should use the command *initdumpfs*(1M) on such computers to create a separate file system containing enough space for memory dumps, i.e. at least 1/6 of total memory.

      If you have not yet created such a file system, then the Init process will output the following warning when the system is initialized:
      DUMPPARTITION isn't set correctly in /etc/default/dumpsave□
      On systems with EHIOS a special dumpdevice (extra mounted file system)□
      must be configured! This will be done by executing once:□
      □□□□□□□□/sbin/initdumpfs

    Please save the *dump.<x>* and *unix.<x>* files and the *info.<x>* archive to an external medium (...), and forward this to the Service department. You should also include a short description of the circumstances in which the system halted.

If a kernel other than *unix* was active when the system halted (e.g. *unix.old*), then this kernel must be included in the error documentation because the core dump cannot be evaluated otherwise.

- If these actions also fail, go into the board debug monitor (...). Disable autoboot there and initiate a cold start:

  RM600 E:

  BDM(SPB□□8 t)>**wf 209**□
  BDM(SPB□□8 t)>**creset**

  RM600-xxx:

  BDM(MBII□□0 t)>**wf 209**□
  BDM(MBII□□0 t)>**creset**

  The autoboot should be disabled so that you can, if necessary, power up the system with a different system kernel, e. g.:

  boot2: **boot -f /unix.old**

  or with the mirror disk or another system disk, e. g.:

  boot2: **root -n <*number*> -s 2**□
  boot2: **boot**

  The *number* variable in the example above stands for the number of the entry of the mirror disk or other system disk in the root partition table. You can look this up with the boot2 command *root -l*.

  When the system has powered up, you should check the file systems for inconsistencies using *fsck*(1M) and view the files */var/adm/log/osm* or */var/adm/log/messages* for hardware error messages.
- If this last action also fails there is presumably a serious hardware or software error. Notify Service in this case.

2.2.5 System bottlenecks

System bottlenecks are indicated by delayed responses to screen inputs, applications taking a long time to respond and, in some cases, warnings on the console.

There can be many reasons for such bottlenecks or drops in performance, some of which are given in the list below.

- File systems are almost full
- Swap space is exhausted
- Memory capacity is exhausted
- Long processes hogging the CPU
- Space in core tables is exhausted
- Too many simultaneous system users
- Performance problems on the application level
- Poor program flow (loops, locks etc.)

Some commands and desktop menus are available in such situations to pinpoint the causes and prepare for remedies:

- Check the utilization of the file systems with the *df*(1) command.
- Check the swap space with the *swap*(1M) command.
- Check accesses to disk with the *dkstat*(1M) command.
- Use the *sar*(1) command to display a sample of CPU utilization, e. g. **sar 6 10** (10 measurements each 6 seconds apart).
- Get an overview of users who are logged in and their processes with the *who*(1), *whodo*(1M) or *finger*(1) commands.
- Use the *ps -ef* command (see *ps*(1)) to see a detailed process list, and scan this for processes that have been running a long time (high figures in the TIME column).

Bottlenecks may arise periodically at certain times. In such cases, it is advisable to observe the situation over a longer period of time. Find out whether any conversions (software or hardware) or changes to the configuration of your system have been carried out since the bottlenecks occurred.

Generally your system should have been configured optimally for its intended usage. Extended usage cannot always be foreseen, however, and can take the system to its performance limits. In this case, it is also necessary to consider expanding capacity or increasing performance, e. g. adding memory or hard disks or installing a more powerful model.

Refer to the "Tuning Guide" for further information in this regard. Our specialists will be pleased to help you with special measurement tools to provide performance analyses and possibly tune the system. Contact Service and have them handle these activities.

3 RM200/300/400 troubleshooting

This chapter is intended to guide you through steps in handling various error situations. This will enable you to get your computer running properly again in the event of minor errors, and to prepare diagnostic documentation for analysis by Service in the event of serious errors.

This chapter is split into three sections.

The section "Overview of systems" contains a table which shows the available diagnostic and monitoring functions.

The "System startup" section explains how you can diagnose errors that occur while the computer is powering up, and resolve them if possible.The overview diagram and the console messages that are shown will help you to establish the phase that your computer is in if any errors occur.

The "System running" section tells you how to diagnose problems that occur while your computer is up and running, and how to solve them if possible. It also explains how to prepare the relevant diagnostic documentation.

All handling instructions relating to the PROM or firmware monitor that are described in this chapter only give the PROM monitor as an example. When using a graphical terminal, the procedure for the firmware monitor is basically the same as shown here. You can find out more about these two monitors in the "PROM monitor" and "Firmware monitor" sections in the "Reliant UNIX Operation" manual.

What else you need to consider:

Have your system ID number (SSY) at hand when reporting problems to Service. It is shown on a gray sticker on the system unit.

Heed the safety instructions in the operating manual for the respective devices (monitor, system unit, auxiliary cabinet, terminal, printer, UPS etc.).

Experience has shown that it is always worth taking a look at the operating and installation instructions for your system if an error occurs. The device- and system-specific advice on troubleshooting that is given in those documents will also be of assistance in solving problems.

3.1 Overview of systems

This section contains a table listing the diagnostic and monitoring functions for all models in the RM400 family which are supported by the Reliant UNIX 5.45 operating system. The following units and associated peripheral cabinets are covered:

- RM300□E
- RM400□E
- RM200□C
- RM300□C
- RM400□C
- RM200□C Rack Unit
- RM300□C Rack Unit
- RM400□C Rack Unit

From Reliant UNIX 5.45A10 the following models are also supported:

- RM200-xxx
- RM400-xxx (Mini Tower)
- RM400-xxx (Tower)

Since these systems do not all have the same range of functions, some of the diagnostic measures suggested are only relevant for a subset of these models. You can check the particular diagnostic and monitoring functions for your system from the tables.

System units

Model	**RM200**			**RM300**		
	-xxx	**C**	**C Rack**	**C**	**C Rack**	**E**
Id of system unit						
Id number begins with	SSY62	SSY66	SSY66	SSY67	SSY67	SSY69
Main memory						
Parity check	•					
EDEC		•	•	•	•	•
Monitoring of						
Temperature□ (internal / environment)				□ •/	□ •/	□ •/•
Fans						•
Supply voltage						•
UPS	•	•	•	•	•	•
Monitoring devices						
I960						•
IDC (option, only for monitoring peripheral cabinets with EIP)				•	•	•
DCU						
SAF-TE						•
Power backup optional with						
Redundant power pack					•	•
UPS	•	•	•	•	•	•
BBU						
Call-up IKDB						
Control panel (Reset key)				•	•	•
Only with key combination	•	•	•			
Console						
Graphics monitor	•	•	•	•	•	•
Alpha terminal	•	•	•	•	•	•

System units (continued)

Model	**RM400**				
	-xxx□ Tower	**-xxx□ Mini□ Tower**	**C**	**C Rack**	**E**
Id of system unit					
Id number begins with	SSY61	SSY60	SSY63	SSY63	SSY68
Main memory					
Parity check	•	•			
EDEC			•	•	•
Monitoring of					
Temperature (internal)	•	•	•	•	•
Fans	•		•	•	•
Supply voltage	•		•	•	•
UPS	•	•	•	•	•
Monitoring devices					
I960					•
EIP	•				
IDC (option, only for monitoring peripheral cabinets with EIP)			•	•	•
DCU			•	•	•
SAF-TE					•
Power backup optional with					
Redundant power supply			•	•	•
UPS	•	•	•	•	•
BBU	•		•	•	•
Call-up IKDB					
Control panel (Reset key)	•	•	•	•	•
Console					
Graphics monitor			•	•	•
Alpha terminal	•	•	•	•	•

Peripheral cabinets

Model	**RM400**		**RM300-BG4x**
	-BG57	**-BG3x**	
Id of peripheral cabinet			
Id number begins with	SBG57		
Can be connected to system unit (You can connect another peripheral cabinet to your system unit by means of migration)			
RM400-xxx Tower	•		
RM300 C, RM400 C		•	•
Monitoring of			
Temperature (internal/environment)	•/-	•/-	•/-
Fans	•	•	•
Supply voltage	•	•	•
Monitoring device			
EIP	•	•	•
Power backup optional with			
Redundant power pack		•	•
UPS	•	•	•
BBU	•	•	
Error indicator			
FAIL (control panel)	•	•	•

Peripheral cabinets (continued)

Model	**OLR□ Hard disks□ Unit Rack**	**DU41**
Id of peripheral cabinet		
Id number begins with		DU40
Can be connected to system unit□ (You can connect another peripheral cabinet to your system unit by means of migration)		
RM300 C Rack	•	
RM400 C Rack	•	
RM300 E		•
RM400 E		•
Monitoring of		
Temperature□ (internal/environment)	□ •/	□ •/•
Fans	•	•
Supply voltage	•	•
Monitoring device		
EIP	•	
SAF-TE		•
Power backup optional with		
Redundant power pack	•	•
UPS	•	•
BBU		
Error indicator		
FAIL (control panel)	•	•

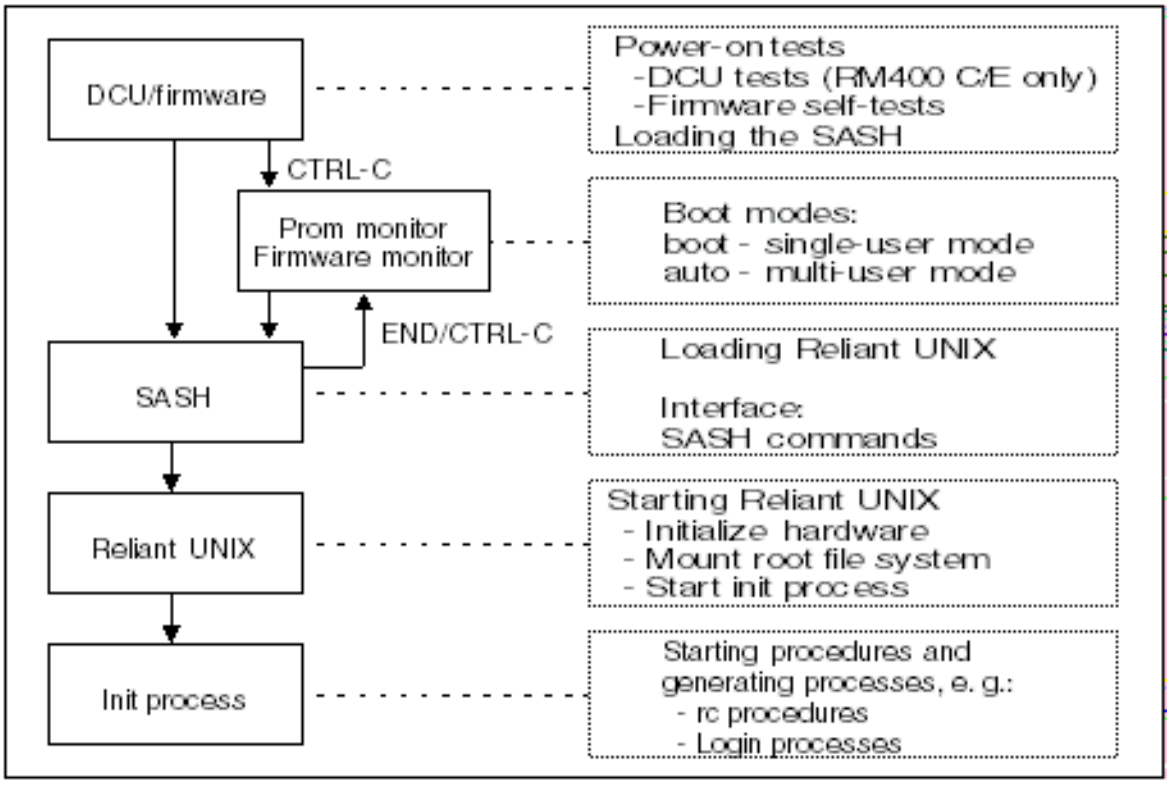
3.2 System startup

This section describes errors that can occur during system startup, how your computer will respond to them as standard, and what action you can take to deal with them.

The entire system startup process is logged on the console. Console messages can often be seen for just a short time because they are scrolled on the screen. To make sure you don't miss any information, you can halt screen output by pressing [CTRL][S] and resume it with [CTRL][Q]. Important console messages during the boot phase, especially kernel and driver messages, are logged in Logging□V3.0.

3.2.1 Overview of system startup

Various control and service programs are started when the system is turned on (see the "Reliant UNIX Installation" manual). An overview of the individual system startup phases up to the beginning of system initialization by the init process is given in the graphic below.



3.2.2 Console messages using an RM400-C72 as an example

3.2.2.1 Power-on tests

DCU tests (RM400 C and E only)

DCU messages only appear when an alpha console is connected.

DCU-Status: Load DCU RAM Version..Passed

DCU-Status: Switch IDC: on/off..Passed

DCU-Status: RAM-Version: 3.6 - 11.04.97

DCU-Status: R10000 Cacheline-Size: 64 Byte

DCU-Status: PLD-Version: 00

DCU-Status: Last Power-on from: Frontpanel

DCU-Status: Last Power-off from: Frontpanel

DCU-Status: Cold Reset from: Power-on

DCU-Status: Start Hardware-Test

DCU-Status: NO BBU found

DCU-Status: External SCSI detected

DCU-Status: Big-Endian Mode

DCU-Status: Read CPU-BOARD-ID: ID-1: 43H

DCU-Status: Check MAUI-DCU-Register

DCU-Status: Check MAUI-DCU-Panic-Register access

DCU-Status: Check Boot-Prom access

DCU-Status: Load Keyboard-Controller....Passed

DCU-Status: IDC found

DCU-Status: Start CPU-Boards

Initializing the memory on a cold start (RM400 C only)

Before the self-test is started the necessary memory areas are first cleared.

RM400_PCI_TOWER Prom Cold-Start:

Version 5.0208 Tue Jun 17 10:24:11 MDT 1997 livsapci

Config. MAUI Memory...PASSED

Copy Prom to Shadow RAM...PASSED

Clear Memory used by PROM(4MB)...PASSED

Read Memory and Skip Exceptions...PASSED

Config. MAUI PCI and Switch Prom...PASSED

Clear Remaining Part of Memory(cached)...PASSED

Firmware self-tests

Firmware messages appear both on an alpha console and on a graphical console. The outputs can vary depending on the computer model and hardware configuration.

Running Power-On Diagnostics...

Processor 01

The following part of the tests lasts several minutes

Cache Test #1...PASSED

Secondary Cache Test...PASSED

ECC Cache Test...PASSED

Data Cache MATS+ Test...PASSED

FP Test #1...PASSED

FP Test #2...PASSED

Processor 0

The following part of the tests lasts several minutes

Cache Test #1...PASSED

Secondary Cache Test...PASSED

ECC Cache Test...PASSED

Data Cache MATS+ Test...PASSED

FP Test #1...PASSED

FP Test #2...PASSED

Memory Test...PASSED

Memory Test...PASSED

Checking the motherboard's SCSI controllers:

SCSI CTLR NCRC810 ONBOARD DEVICE 01 :

...

IdProm Test...PASSED

□□Ethernet address: 08:00:06:08:4e:3d

□□Ethernet Checksum: 0x03

LAN AMD79C970 CONTROLLER : 00

□□Ethernet Address...PASSED

□□Port test...PASSED

□□Lan Internal LoopBack Test...PASSED

□□Lan External LoopBack Test...PASSED

Interrupt controller for the EISA bus:

EISA ICU Test...PASSED

Serial interface:

Duart 452/552 Port Tests...Channel B...PASSED

Parallel printer interface:

Centronics Port Test...PASSED

8254 Timer 00 Test...PASSED

8254 Timer 02 Test...PASSED

Time-of-Day Clock Test...PASSED

Ending Power-On Diagnostics...

Machine is coming up...

3.2.2.2 Firmware Monitor

The following will be output if the firmware environment variable *bootmode* has some value other than "C", or if you press [CTRL][C] to abort automatic loading of the SASH. It shows you the memory configuration of your system and enables you to recognize whether any components have been deactivated by the LAR facility.

SNI Monitor Version 6.0003 Mon Nov 30 18:14:13 MET 1998 livsapci

Multiprocessor System with 4 processor(s)

ASIC (MAUI) Revision 2

Memory size: 576 MBytes (0x24000000 bytes)

Modul#1: 256MB at 0x020000000 Modul#2: 256MB at 0x030000000

Modul#3: 64MB at 0x040000000 Modul#4: 0MB at 0x00

Modul#5: 0MB at 0x00 Modul#6: 0MB at 0x00

Modul#7: 0MB at 0x00 Modul#8: 0MB at 0x00

LAR (Logout Auto Recovery) Mode enabled

No CPU or memory modules are set defect

CPU : R10000 SC, Revision 3.4

Primary I-cache size: 32768 (0x8000) bytes

Primary D-cache size: 32768 (0x8000) bytes

Secondary cache size: 4096 Kbytes (0x400000 bytes)

3.2.2.3 Loading the SASH

You can press [CTRL][C] at this juncture in order to interrupt the automatic loading process and, for instance, load the Mini System, execute PROM monitor commands or boot Reliant UNIX in single-user mode.

Autoboot: Waiting to load dkncr(0,0,10)sash (CTRL-C to abort, RETURN to

expedite)

loading

407596+0+557060 entry: 0xa0e00000

SNI Standalone Shell Version 5.0208 Tue Jun 17 10:24:35 MDT 1997 livsapci

3.2.2.4 Loading the Reliant UNIX kernel

Loading dkncr(0,0,0)unix

3868004+2189328+1406376 entry: 0x80030100

SINIX-N: HAL (pcit) 97/03/10 14:11:08

Board Type: 6

SNI RM400

3.2.2.5 Starting the Reliant UNIX kernel

The outputs can vary depending on the computer model and hardware configuration.

Initializing the hardware and mounting the root file system

UNIX(R) SINIX-N Release 5.45 Version A0042

Copyright (c) Siemens AG 1998

Basis:□DC/OSx (R), Copyright (c) Siemens Pyramid Information Systems, Inc. 1984;

□□□□□□□UNIX (R), Copyright (c) The Open Group 1983

All rights reserved

SINIX is a registered trademark of Siemens AG

DC/OSx is a registered trademark of Siemens Pyramid Information Systems, Inc.

Reliant is a registered trademark of Siemens AG

X/Open is a registered trademark, and the X device and The Open Group

are trademarks, of The Open Group in the US and other countries.

UNIX is a registered trademark of The Open Group

Processor Configuration:

Boot processor: Cpu #0 (p0)

Other processor(s): #1 (p1), #2 (p2), #3 (p3)

CPU 0: MIPS R10000 Processor Chip Revision: 3.4

FPU 0: MIPS R10000 Floating Point Unit Revision: 0.0

Memory Map:

□□□□□□□□□□□□□□□□0x0000000000000000 - 0x00000000ffffffff

□□□□□□□□□□□□□□□□0x0000000030000000 - 0x0000000043ffffff

Delay multiplier = 7998, cnt = 9547

CPU Freq = 250 MHZ

□debug : mode <3>, logging not defined

CMN: NOTICE: cmn 46018 bfreelist.b_bufsize = 0x5c4000

Configuring dfe to use interrupt 10

PCIT: 82374 EISA SYSTEM COMPONENT: REVISION ID = 0x13

CMN: NOTICE: cmn 36354 alt5: Using Local MAC address 0:60:cf:20:11:c2

Mirror Disk initialization ... completed.

ReliantUNIX enhanced Networking

□□□□□(c) 1986,1987,1988,1989 Sun Microsystems, Inc.

□□□□□(c) 1983,1984,1985,1986,1987,1988,1989 AT&T.

□□□□□(c) 1987, 1988, 1989 Lachman Associates, Incorporated (LAI)

□□□□□(c) 1982, 1986, 1988, 1993 The Regents of the University of California.

□□□□□(c) 1998 Siemens AG.

CMN: NOTICE: cmn 18838 alt5: Serial Number is 0:60:cf:20:11:c2

CMN: NOTICE: cmn 35849 alt5: Part number is 200007P2A

CMN: NOTICE: cmn 6106 alt5: Board Revision is P1

CMN: NOTICE: cmn 36354 alt5: Using Local MAC address 0:60:cf:20:11:c2

-- cpumap = 0f : larcpu = 00

cpubrd00 : cpu board configuration

-- cpumap = 0f : larcpu = 00

cpubrd01 : cpu board configuration

[90000000140003bc] cpt00 : COMMDEV SNI VL16C552 (CENTRONICS port)

[90000000140003F8] du01 : VL16C552 : (Port 0: enabled) (Port 1: enabled)

[90000000140003E8] du03 : VL16C552 : (Port 2: enabled) (Port 3: enabled)

Controller detected on the EISA bus (some of these components are on the motherboard):

eisa : EISA-BUS configuration

[140003F2] fd01 : FLOPPY SNI i82077AA (floppy disk cntr.)

CMN: NOTICE: cmn 7317 alt5: Gigabit Ethernet link is negotiated

CMN: NOTICE: cmn 44121 alt5: rx flow control is enabled, tx flow control is

enabled

dfe: Initializing DEC FDDIcontroller/EISA (dfe0)...

[9000000014001000] dfe02 FDDI DEC DEFEA 3001 [08-00-2B-A0-76-03], slot=EISA#1

Controller detected on the PCI bus (some of these components are on the motherboard):

pci : PCI-BUS configuration

dfe: Initializing DEC FDDIcontroller/PCI (dfe1)...

[9000000018314000] dfe01 FDDI DEC DEFPA [00-00-F8-C8-AE-01], slot=PCI#4

sport01: 0 1 8

[1400B090] ios0/scon01 : SCSI DPT PM3334 SE, slot=PCI#3/0, FW 07LP 4MB

emc0: I960 Server Management, I2O-IOP#0 (IopID:0, Ver:34) PCI#ONBOARD

[90000000140003C0] gfx00 : DISPLAY SNI CIRRUS 5436 (Console, 2 MB), slot=PCI#9

[9000000014009000] madge00 TOKEN MADGE Smart 16/4 [00-00-F6-B5-00-AD], CL32,

slot=PCI#11 16Mb

[900000001400a000] madge01 TOKEN MADGE Smart 16/4 [00-00-F6-B5-71-32], CL32,

slot=PCI#8 16Mb

[9000000018101100] ios0/scon00 : SCSI SNI SYM53C875 SE, slot=PCI#ONBOARD

[18101200] zx00 ETHER ZNYX 21140 V34 [08-00-06-0D-84-3B] 40.1, slot=PCI#ONBOARD

[18000180] zx10 ETHER ZNYX 21140 V34 [00-C0-95-E0-1A-48] 90.3, slot=PCI#1/4

[18000100] zx11 ETHER ZNYX 21140 V34 [00-C0-95-E0-1A-49] 90.3, slot=PCI#1/5

[18000080] zx12 ETHER ZNYX 21140 V34 [00-C0-95-E0-1A-4A] 90.3, slot=PCI#1/6

[18000000] zx13 ETHER ZNYX 21140 V34 [00-C0-95-E0-1A-4B] 90.3, slot=PCI#1/7

Motherboard and CPU configuration:

-- mbrd_type = 0x0a, mbrd_module = 00339.00.4.15

-- mbrd_serial = 000000006B00457654, mbrd_revision = 0x80 0x00 0x00 0x00 ...

-- cpubrd1_type = 0x30, cpubrd1_module = .

-- cpubrd1_serial = 000000000000000000, cpubrd1_layout = 30,

cpubrd1_revision = 0x00 0x00 0x00 0x00 0x00 0x00

-- frequency = 250

-- frequency = 250

-- cpubrd0_type = 0xff, cpubrd0_module = 08729.00.3.12

-- cpubrd0_serial = 000000006B00457654, cpubrd0_revision = 0x80 0x00 0x00 ...

-- frequency = 250

-- frequency = 250

-- larmem[B8,..,B1] =0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00

-- prom version = 6.0003

-- ponmask = 0x00000000, failcode=0x00, state=0x03

-- resetepc= 0xf439c1bf, resetra=0xf406c0bf

-- service_id = SSY6804711

-- eip_chan = 0x00, eisa_ok = 0x00

>> mod_name = RM400-E60, cab_type = SY68, cab_name = PCI-Tower

SINIX-N: HAL (pcit) 1.22.1.2 98/01/16 09:21:23

Board Type: 10

Main memory capacity:

Slot #1: 256 MBytes Slot #2: 256 Mbytes

Slot #3: 64 MBytes Slot #4: 0 Mbytes

Slot #5: 0 MBytes Slot #6: 0 Mbytes

Slot #7: 0 MBytes Slot #8: 0 Mbytes

Secondary Cache (I + D mixed) = 4194304

-- System coming up: SINIX-N 5.45 A0042 (MP)

Total real memory = 576 MB

Available memory = 548 MB

-- BootCPU <lcpu,pcpu>: <0,0>

-- Starting <lcpu,pcpu>: <1,1> <2,2> <3,3>

System initialization

Detection of console type and allocation of the console device node:

sport00: 5

CMN: NOTICE: conclone 2 set_maj_minor:

Real console device switched to (128,5)

Consistency check of the root file system:

The root file system (/dev/rroot) appears clean.

link /dev/rroot to /dev/ios0/rsdisk010s0

link /dev/root to /dev/ios0/sdisk010s0

Node: atlantis

Configuration of virtual disks (if any):

/sbin/dkconfig -aNc

/etc/dktab doesn't contain non-grouped vdisk declarations

The system is coming up. Please wait.

Start of rc procedures:

rc2 start:

rc2: /etc/rc2.d/S00acls start

rc2: /etc/rc2.d/S00chid start

Checking channel_id configuration ... done

Compare configuration in NVRAM with the entry on the hard disk:

rc2: /etc/rc2.d/S01MOUNT0FSYS start

Check /var, /opt, /usr, /home:

Checking Level 0 Filesystems...

Mount swap area:

rc2: /etc/rc2.d/S03swap start

link /dev/swap to /dev/ios0/sdisk010s1

rc2: /etc/rc2.d/S04MOUNT1FSYS start

Checking Levels 1-9 filesystems only...

rc2: /etc/rc2.d/S05RMTMPFILES start

rc2: /etc/rc2.d/S05emc start

rc2: /etc/rc2.d/S06log3 start

Starting the logging daemon ...

Started!

Start the syslog daemon ...

Started!

rc2: /etc/rc2.d/S07lar start

/usr/bin/lar/update_scsi ... done

/usr/bin/lar/updmemcpu ... done

rc2: /etc/rc2.d/S07restore start

rc2: /etc/rc2.d/S08saveinfo start

rc2: /etc/rc2.d/S09osm start

osmd daemon started!

rc2: /etc/rc2.d/S10ultra start

rc2: /etc/rc2.d/S11ullc start

rc2: /etc/rc2.d/S12alt start

rc2: /etc/rc2.d/S12dfe start

rc2: /etc/rc2.d/S12et start

rc2: /etc/rc2.d/S12madge start

rc2: /etc/rc2.d/S12pnet start

rc2: /etc/rc2.d/S12zx start

rc2: /etc/rc2.d/S13op_disk start

Checking device nodes:

rc2: /etc/rc2.d/S13scanports start

Check for SCSI device location done

rc2: /etc/rc2.d/S14migrate_devices start

rc2: /etc/rc2.d/S14saveautoc start

rc2: /etc/rc2.d/S15mkdtab start

rc2: /etc/rc2.d/S18silsd start

Siemens License Daemon started

rc2: /etc/rc2.d/S19tt_open start

rc2: /etc/rc2.d/S20sysetup start

rc2: /etc/rc2.d/S21perf start

rc2: /etc/rc2.d/S21timeshare start

rc2: /etc/rc2.d/S30nrd start

INFO: name root disk ...

INFO: ios0/sdisk010 got name "MP38: 5.45 A0042 "

rc2: /etc/rc2.d/S40safte start

rc2: /etc/rc2.d/S51uts start

rc2: /etc/rc2.d/S65netconf start

Ethernet seems to be configured.

rc2: /etc/rc2.d/S69PREinet start

rc2: /etc/rc2.d/S69inet start

add net default: gateway 139.25.22.94

add net 139.25.22.128: gateway 139.25.22.41

rc2: /etc/rc2.d/S69syslog start

syslogd udp-service activated by signal SIGUSR2

rc2: /etc/rc2.d/S70SIlvs start

Starting lvs

rc2: /etc/rc2.d/S70termadm start

rc2: /etc/rc2.d/S70uucp start

rc2: /etc/rc2.d/S71SAFEchk start

rc2: /etc/rc2.d/S72eecd start

rc2: /etc/rc2.d/S75cron start

rc2: /etc/rc2.d/S75rpc start

rc2: /etc/rc2.d/S76yp start

rc2: /etc/rc2.d/S77snmpdm start

SNMP Research SNMP Agent Resident Module Version 14.2.1.18

Copyright 1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997 SNMP Research,

Inc.

rc2: /etc/rc2.d/S78snmpd start

Sinix SNMP Agent Adapter Resident Module Version 3.0

Copyright 1989, 1990 SNMP Research, Inc.

Copyright (C) Siemens AG 1998

All rights reserved

rc2: /etc/rc2.d/S83lsm start

rc2: /etc/rc2.d/S88smtpd start

rc2: /etc/rc2.d/S92tcpqa start

rc2: /etc/rc2.d/S92tcpqad start

Starting UDP unamed...

rc2: /etc/rc2.d/S95nfs start

rc2: /etc/rc2.d/S95ports start

rc2: /etc/rc2.d/S96dpt start

DPT 29.01 08:54 DPT Event Logger Loaded

rc2: /etc/rc2.d/S96memsz start

rc2: /etc/rc2.d/S96sandm start

/etc/rc2.d/S96sandm

NOTICE: sandm 2 Found 1024288 KB of swap space, which is

□□□□□□□□□□□□□□□□□□□□□□□□□enough for 589824 KB of main memory

rc2: /etc/rc2.d/S97Slapache start

/etc/rc2.d/S97Slapache start: httpd started

rc2: /etc/rc2.d/S99conf_ana start

rc2: /etc/rc2.d/S99mocd start

rc2: /etc/rc2.d/S99motif start

rc2: /etc/rc2.d/S99saf start

tty monitor already created.

rc2: /etc/rc2.d/S99updateguard start

rc2: /etc/rc2.d/S99z_last start

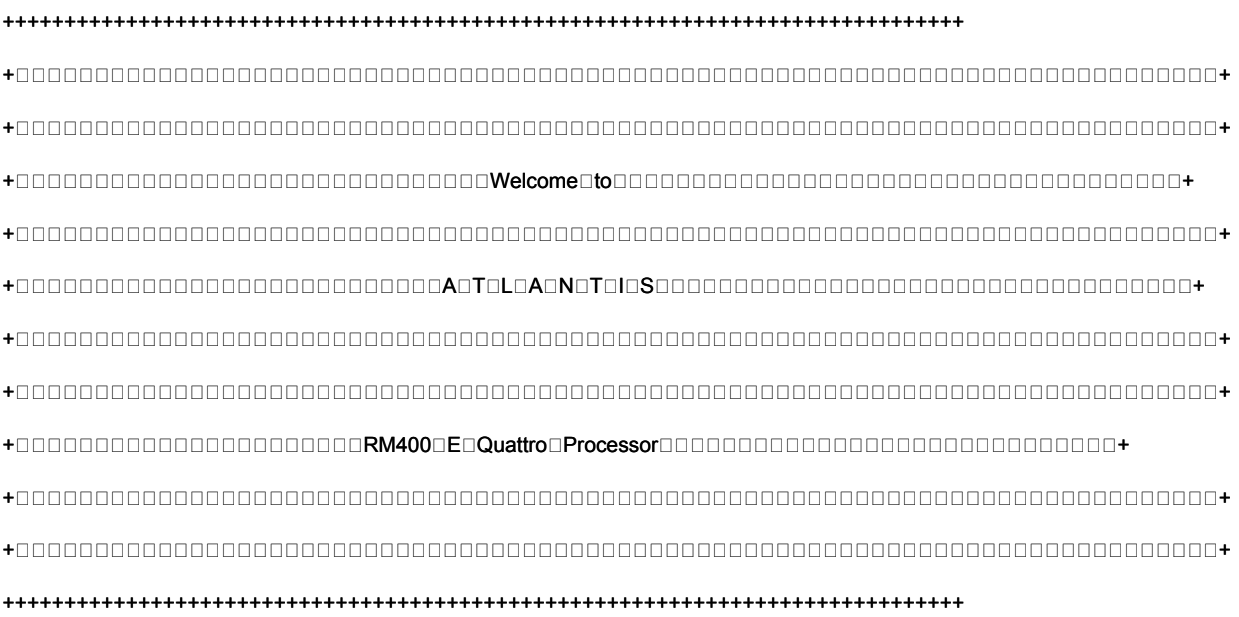
done.

Mobo: Memory:0:slot:1: error set by system! Cause: PCITHW 21: osm: CMN:

WARNING: PCITHW 21 MM corrected at 0x42740980

The system is ready.

3.2.2.6 Output of /etc/issue and login



Console Login:

When a graphical console is installed, the X server is also started now and the SINIX/windows graphical user interface appears with a login form.

You can use the following key combination on graphical consoles to switch from graphical mode to alpha mode:

Press and hold down the [ALT] key, press the [SysRq] key (international keyboard) or the [S-Abf] key (German keyboard), release both keys and then press the [H] key.

You can use the following key on graphical consoles to switch from alpha mode to graphical mode:

Press and hold down the [ALT] key, press the [SysRq] key or the [S-Abf] key, release both keys and then press the [F1] key.

3.2.3 Turning on the system

The following problems can occur when the system is turned on. Contact Service if a suggested approach does not solve a problem.

3.2.3.1 System cabinet

- The *POWER* LED on the system cabinet does not light up.

  No power is being supplied or the system was not turned on properly (keyswitch not on LOCAL).
  - Turn the system on again or check the power supply.

  RM200 C, RM300 C: The power pack has not been turned on.
  - Turn the power pack on by setting the switch on the back of the housing to I.

- The system unit does not switch on automatically following a power failure.

  A wait time for recharging the batteries in the UPS is set in the system.
  - You can use the *ups*(1M) and the *pwr*(8) command to define the behavior following a power failure when you are using a UPS.

- An RM400-C80 or C90 switches off again a few seconds after power-on. If you are using a serial console, you will see the following message:
  *Error: ACFAIL_N from Power-Supply□
  DCU: Switch Power off in 2 sec!!!

  Only one of two power cables is connected.
  - Check the power supply.

- The system cannot be turned on remotely (The RM400-10 and RM200-xxx systems cannot be turned on remotely.).

  No power is being supplied.
  - Check the power supply.

  The keyswitch on the central unit of the remote system has not been set to REMOTE (RM300 C/400 C).
  - Turn the keyswitch to REMOTE.

  The remote device (terminal, printer) or the communication device (e. g. modem) has not been turned on.
  - Turn the remote device or communication device on.

  The remote device is not connected correctly.
  - Check the connections from the system to the remote device.

For RM400-XXX (Tower) only:

- Messages are output on the LCD control panel following power-on. You will find an explanation of these messages in the operating manual. The following errors may be reported:

  <<POWER FAIL>>
  AUTO OFF:xxxxsec

  The power has failed.
  - Check the power supply.

  <<TEMP FAIL>>
  AUTO OFF:xxxxsec

  The temperature in the system unit has risen to above 40° C.
  - Check if the ambient temperature is above 32° C and make sure the air vents are not obstructed. The high temperature may also be caused by a fan failure. To establish this, cool down the system unit and see if you get a "FAN FAIL" when you power on the system again.

  <<FAN FAIL>>
  AUTO OFF:xxxxsec

  A fan has failed in the system.
  - Contact Service.

3.2.3.2 Peripheral cabinets

- The peripheral cabinet does not switch on with the system. The *ON/BATT* LED does not light up.

  No power is being supplied.
  - Check the power supply, and start up the system again.

  The control line from the system cabinet to the CAN/EIP bus of the peripheral cabinet is disrupted.

  Communication between the EIPs (Environmental Interface Processor) or IDC (ISA diagnostic bus controller) in the peripheral and system cabinets is conducted by means of the CAN bus. In addition to environment monitoring data, the commands for turning on and off the power packs are also handled via this interface.
  - Check the peripheral cabinet cabling, and start up the system again.

  The EIP has not been configured correctly.
  - Have Service check the EIP configuration.

- The peripheral cabinet only switches on briefly.

  There is a hardware error.
  - Look at the error messages on the console and notify Service.

3.2.3.3 Peripheral boxes

- The peripheral box does not switch on with the system. The *ON/BATT* LED does not light up.

No power is being supplied.

- Check the power supply, and start up the system again.

The power pack of the peripheral box has not been turned on (the power pack has an on/off switch, depending on the model).

- Turn the power pack on by setting the switch to I, and start up the system again.

The control line from the system cabinet to the SCSI bus of the peripheral box is disrupted.

- Check the peripheral box cabling, and start up the system again.

- The peripheral box does not switch off automatically.

  The switch on the back is set to LOCAL.
  - Set the switch to REMOTE.

3.2.3.4 Console

- The screen remains dark after switching on.

  No power is being supplied.
  - Check the power supply.

  The screen has not been turned on.
  - Turn the screen on.

    With some graphics monitors, the screen only lights up approx. 20 seconds after the monitor is switched on.

  The screen was dimmed automatically after a long period of inactivity.
  - Press any key.

  The brightness control is not set correctly.
  - Adjust the brightness (see the operating instructions for your monitor).

  The cable for the VGA monitor is connected to the standard port although a PCI graphics controller has been installed.
  - Connect the monitor to the PCI graphics controller (see the operating instructions for your computer).

The graphics monitor type specified at startup is not the same as the monitor now connected. You can switch to VGA-ANSI mode, which is supported by all graphics monitors, and set the new type.

- Press and hold down the [ALT] key, press the [S-Abf] (German keyboard) or [SysRq] (international keyboard) key, release both keys and then press the [H] key.

  You are now in VGA-ANSI mode. Invoke the *sysadm* command and select the menu item *console_config — xserver Hardware — Monitor Type*. The correct type is now entered. To see the list of supported models, press the CHOICES softkey.

  The new setting is active the next time the system is booted.

The VGA monitor is connected via the 5 BNC connectors and the selected type of input is D-SUB, or vice versa (see the operating instructions for your monitor).

- Check the cabling on the back of the monitor. Define a different type of connection either by changing the type of input on the front control panel of your monitor (INPUT SEL or MODE) or by changing the cabling of your monitor.

There is a system hardware error.

- Make a note of the status of the LEDs on the back of the system cabinet (RM400 C only) and notify Service.

- The screen is light, the cursor is flashing but no input is possible.

  The monitor is not connected to the computer.

  - Check whether the plugs at the end of the cable have been inserted in the sockets correctly.

  The central unit has not switched on or is blocked.

  - Check the central unit (see the section System cabinet).

- The screen (alpha monitor) does not show any characters, or just garbled ones.

  The screen parameters have not been set correctly.
  - Set the screen parameters correctly (see the operating manual for your monitor). The following default values have been set in the system unit for the console:
    Baud rate: 19200 bits/s
    Parity: None
    Character length: 8-bit
    Number of stop bits: 1

- The operating parameters that have been set for a TC20 monitor are lost.

  The setup/main menu was exited without saving the settings.
  - Call the setup/main menu again and select *save menu settings*.

- For malfunctions relating to the UPS cabinet, consult the operating instructions that are supplied with it.

3.2.4 Power-on tests

Power-on tests are started when the system has been turned on (see the "Reliant UNIX Operation" manual for more details). Device Control Unit (DCU) messages only appear on an RM400 C or RM400 E to which an alpha console is connected. Messages from the firmware self-tests are output on a graphical console and on an alpha console.

The following problems can occur during the power-on tests. Notify Service if a suggested approach does not solve a problem

3.2.4.1 DCU tests (RM400 C and E only)

- The DCU turns the system cabinet off and outputs an error if an alpha console is connected. A hardware error has occurred, e. g.:
  DCU-Status: DCU-Hardware-Test□
  *Error: Thermo-Click CPU-Board-1□
  *Error: MAUI-Register Access Hardware-Error□
  DCU: Switch Power off in 2 sec!!
  - Make a note of the error message and turn the system on again.

- An RM400-C80 or C90 switches off again a few seconds after power-on.

  Only one of two power cables is connected. If you are using a serial console, you will see the following message:
  *Error: ACFAIL_N from Power-Supply□
  DCU: Switch Power off in 2 sec!!!
  - Check the power supply.

3.2.4.2 Firmware self-tests

- If a test in this phase does not run without error, the FAILED status is output instead of PASSED. The boot mode is automatically set to *e* to prevent Reliant UNIX from being loaded. The firmware self-tests are terminated with the following message:
  *****□□□Problems on 0x???? Motherboard

  An error has been propagated from an external connection, e. g. LAN, SCSI, serial or parallel port.
  - Remove the connection and turn your system on again.

  A hardware error has occurred in the system.
  - Make a note of the error message and turn the system on again.

- If the system does not power up automatically and remains in the PROM monitor without outputting an error message, then there can be the following reasons:

  The boot mode has been set to *e* because the system detected an error previously, or the boot mode has been set to *d* for diagnostic purposes.
  - Set the boot mode to *c* and continue system startup.

```
>> setenv bootmode c
>> auto
```

The boot mode has been set to *d* via a DIP switch on the motherboard.

- Notify Service to correct the setting.

The boot mode has been set to *c* on an RM200 C with an alpha console (server version). The boot mode must be *s* (server) for this configuration.

- Set the boot mode to *s* and continue system startup.

```
>> setenv bootmode s
>> auto
```

3.2.4.3 Other problems

- There can be the following reasons for the power-on tests taking an unusually long time and possibly not completing:

  The tests can take up to 20 minutes, depending on the hardware configuration. This is normal behavior.

  A motherboard component is defective and is blocking the tests.
  - Make a note of the last test performed, and turn the system on again.

- You see one of the following messages on the graphical console:

  no keyboard connected

  The keyboard has not been connected, or the keyboard and mouse connections have been reversed.
  - Connect the keyboard or put the connections right, and turn the system on again.

  no keyboard and no mouse connected

  The keyboard and mouse have not been connected.
  - Connect the keyboard and mouse, and turn the system on again.

  no mouse connected

  The mouse is not connected to the mouse port.
  - Connect the mouse. The correct port is given in the operating manual for your system.

- The following message is displayed on the console, and the system goes into the PROM monitor.
  NVRAM: CHECKSUM ERROR□
  >>

  The contents of the NVRAM have been corrupted (e. g. through loss of voltage), or the motherboard has been replaced and the NVRAM does not contain any data or contains incorrect data.

  - Initialize the NVRAM with the following firmware command:

    >> **init_e2prom**

    You will now see a list of parameters from the NVRAM. You can ignore the values. To end the interactive command, select *save*:

    Command ((q)uit, (e)dit, (s)ave or exit)? **s**

    Provided the NVRAM was deleted, it is now assigned default values and saved with a checksum.
  - Set the boot mode to *c* or *s* (RM200 server version) and continue system startup:

    >> **setenv bootmode c**□
    >> **setenv bootfile dkncr(0,0,10)sash**□
    >> **auto**

    
    This setting applies for standard configurations. Boot from an alternative disk or from a RAID array, so that you can output the bootstring with the firmware command *dkprint*.

  If the system still won't boot, this is not the correct parameter for *bootfile*, or a hardware error has occurred.

  - Notify Service in this case.

3.2.5 Loading the standalone shell (SASH)

When the power-on tests have been completed successfully and the right boot mode (see the section Firmware self-tests) has been set, the firmware loads the SASH.

The SASH and Reliant UNIX are loaded from the disk that is specified in the firmware variable bootfile. By default this is the system disk that is connected to the first SCSI controller and has SCSI ID 0. It is possible, however, to load the SASH and Reliant UNIX from another hard disk. See the "Reliant UNIX Operation" manual for more details.

The following problems can occur when the SASH is being loaded. Notify Service if a suggested approach does not solve a problem.

- The SASH cannot be loaded because the boot disk cannot be accessed. The following message is displayed on the console:
  dkncr(0,0,10): failed inquiry; Selection or Reselection Timeout occurred

  The disk has not been connected properly or is defective.
  - Check the hardware and turn your system on again. Notify Service in any case.
- The SASH cannot be found by the firmware. The system automatically goes into the PROM or firmware monitor.

  The disk specified in the firmware variable *bootfile* is not a boot disk.
  - Check the variable, put it right, and continue system startup:

    >> printenv bootfile
    >> setenv bootfile ...
    >> auto

  The SASH does not exist in partition 10 on the disk, or has been corrupted.
  - Load the Mini System (...), and then load the SASH from CD to the system disk using the following option on the menu:

    Write SASH to a System Disk

    When the SASH has been written to the system disk, exit the Mini System via the following option on the menu:

    Abort Mini System and Reboot

3.2.6 Loading Reliant UNIX

When the SASH has been loaded successfully, it in turn loads the Reliant UNIX system kernel into memory (see the "Reliant UNIX Operation" manual for more details).

The following problems can occur when Reliant UNIX is being loaded. Notify Service if a suggested approach does not solve a problem.

- Loading is aborted with the following error message, and you see the SASH prompt:
  dkncr(0,0,10)sash: text section would overwrite sash text, data, bss or stack
  sash:

  When SASH and Reliant UNIX are loaded, a common load area of a certain size is assigned to the two of them. When Reliant UNIX was being loaded, an attempt was made to overwrite the SASH that is in memory, and SASH prevented this.
  - The Reliant UNIX kernel is too large. Load another kernel, e. g. *unix.old* (...), so that you can start the system again.

    The size of the kernel depends, among other things, on the number of components available in the new system kernel for controllers, special operating system functions and modified tuning parameters. You can always generate a smaller kernel for your system by removing software packages for controllers that are not needed. You should also refer here to the information given on system tuning (see the

"Reliant UNIX 5.44 - Tuning Guide").

You tried to load Reliant UNIX using the firmware command *auto* from the SASH.

- Press [CTRL][C] to go into the PROM monitor and continue system startup:

  >> **auto**

- Loading is aborted with the following error message, and you see the SASH prompt:
  unix not found, couldn't load dkncr(0,0,0)unix□
  sash:

  The *unix* file cannot be found in the root partition of the boot disk because the root file system was not terminated correctly the last time it ran or because the *unix* file was deleted.
  - Load the Mini System (...). Go to a shell from the main menu of the Mini System and check the root file system (The root system is on */dev/ios0/sdisk000s0* by default. If your root file system is on another boot disk, you have to specify that.):

    fsck -F ufs -l /dev/ios0/sdisk000s0

    If the root file system cannot be restored, or if data is lost in the process, you should restore the last system backup.

    If the root file system does not contain errors or has been restored, exit the Mini System via the following menu item:

    Abort Mini System and Reboot

  - If loading is aborted again with the same message, load another system kernel, e. g. *unix.old* (...).
- Loading hangs without an error message.

  The Reliant UNIX kernel has been corrupted.
  - RM300/400:
    Press the RESET button for more than 3 seconds (keyswitch on LOCAL) and load another system kernel (e. g. *unix.old*).

    RM200:
    Turn the system on again and load another system kernel (e. g. *unix.old*).

3.2.7 Starting Reliant UNIX

Once the Reliant UNIX system kernel has been successfully loaded into memory, Reliant UNIX is started (see the "Reliant UNIX Operation" manual for more details). In this phase, among other things, the hardware is initialized, the root file system is mounted and the system is initialized by the init process.

3.2.7.1 Initializing the hardware and mounting the root file system

The following problems can occur in this phase. Notify Service if a suggested approach does not solve a problem.

- Hardware initialization is aborted, and control passes to the kernel debugger *ikdb*.

  An interrupt conflict has occurred.
  - Generate a core dump for diagnostic purposes (see the section System halts).
  - Load another system kernel, e. g. *unix.old* (...), so that you can use your system again.

- The startup process is aborted with the following error message:
  no kernel virtual space.size=<*xx*>, base=<*xx*>

  The Reliant UNIX kernel has not been configured correctly for your hardware.
  - RM300/400:
    Press the RESET button for more than 3 seconds (keyswitch on LOCAL) and load another system kernel (e. g. *unix.old*).

    RM200:
    Turn the system on again and load another system kernel (e. g. *unix.old*).

  A hardware problem has occurred in the memory environment.
  - Notify Service.

- The system hangs during hardware initialization without an error message.

  A driver cannot initialize the relevant hardware because, for example, an interrupt conflict has occurred or the board has not been installed.

  - RM300/400:
    Press the RESET button for more than 3 seconds (keyswitch on LOCAL) and load another system kernel, e. g. *unix.old* (...).

    RM200:
    Turn the system on again and load another system kernel (e. g. *unix.old*).

  The root file system may have been corrupted.

  - Load the Mini System (...). Go to a shell from the main menu of the Mini System and check the root file system, e. g.:

    fsck -F ufs -l /dev/ios0/sdisk000s0

    If the root file system cannot be restored, or if data is lost in the process, you should restore the last system backup.

    If the root file system does not contain errors or has been restored, exit the Mini System via the following menu item:

    Abort Mini System and Reboot

- The startup process is aborted with the following error message:
  cannot mount root-Filesystem

  The root file system has been corrupted.

  - Proceed as described above under "The root file system may have been corrupted".

3.2.7.2 System initialization with the init process

When the hardware has been initialized and the root file system has been mounted, the first process to be called is the init process. The main functions of this process are to generate further processes from the entries in the */etc/inittab*(4) file, to start procedures (e. g. *rc* procedures, see *rc2*(1M)) and to start login services for the console and local connections. (see the "Reliant UNIX Operation" manual for more details of system initialization)

Problems during system initialization fall into three categories:

- Incorrect initialization steps without aborting system startup

  In such cases, you have to be prepared for individual components to malfunction when the system is up and running.
- Hangup situations which do not resolve themselves
- Incorrect initialization steps with system startup aborting

Problem analysis is mainly confined to evaluating "suspicious" messages and pinpointing the time of the error. Most attention should be paid to those error messages which are displayed immediately before the error event.

These messages should be jotted down if possible. This is especially true if system startup is aborted or a hangup situation occurs so there is no access to the system and its log files.

During the initialization phase it is often impossible to localize error messages precisely because the screen is scrolled quickly and many messages are output. Furthermore, error messages can often barely be distinguished from status information. In these circumstances it is advisable to evaluate the log files by means of Logging□V3.0 or the diagnosis menu. Additional evaluation resources are available if required (...).

The following problems can occur in the initialization phase. Notify Service if a suggested approach does not solve a problem.

Incorrect initialization steps without aborting system startup

- Errors are reported from *rc* procedures.

  The hardware or software has been configured incorrectly, or not at all.
  - Make a note of the error messages and evaluate the log files. Find the errors and correct them.

  The hardware is malfunctioning.
  - Check whether all devices have been turned on and connected correctly.

- You see the following error message on the console:
  INIT: Command is respawning too rapidly. Check for possible errors.
  id: *</etc/inittab_entry>*

  The entry referred to in the */etc/inittab* file has not been executed properly on more than one occasion (e. g. because of a malfunctioning input/output interface).
  - Check the entry in the file and eliminate the cause of the error message. Then check the entire file using the *init q* command.

- The following message is displayed on the console:
  <*rc_procedure*> failed with exit status of 100.
  SYSTEM is going to SINGLE USER.
  Check the script and restart.

  A serious error has occurred in the stated *rc* procedure. The system is powered up in single-user mode, and you must enter the *root* password to get a shell.
  Type Ctrl-d to proceed with normal startup,
  (or give root password for system maintenance):
  - Check the bad *rc* procedure, possibly together with Service, eliminate the cause of the error and power up your system again with *init 6*.
  - To be able to work with the system on a restricted basis at least, you can also try to power up the system in multi-user mode without this script. To do this, rename the file and power up your system again:

    mv /etc/rc2.d/S69inet /etc/rc2.d/s69inet
    init 6

Hangup situations which do not resolve themselves

- System startup hangs during the initialization phase without an error message.

  There is a hardware malfunction, or the software is inconsistent, buggy or configured incorrectly, or an individual process that was started from an *rc* procedure is hanging.

  - RM300/400:
    Take note of the previous messages and terminate the system via the kernel debugger: Press the RESET button for less than 3 seconds and enter the following commands in the given order:

    ```
    ikdb> pa
    ikdb> ex
    ikdb> ex
    ```

    Then attempt to boot your system again.
  - RM200:
    Turn on the system again.

- System startup is halted with the following message at the start of the init process:
  ENTER RUN LEVEL(0-6,s or S):

  The *initdefault* entry is missing from the file */etc/inittab*.

  - Specify a valid operating state (generally **2** for multi-user mode) and, when the system has powered up, insert the *initdefault* entry at the following position in the file, e. g.:

    ```
    brc:1234:bootwait:/sbin/brc >/dev/console m2>&1 </dev/console
    is:2:initdefault:
    rS:S:wait:/sbin/rcS >/dev/console 2>&1 </dev/console
    ```

Incorrect initialization steps with system startup aborting

- The system is automatically rebooted when the root file system is checked:
  The root file system (<*file_system*>) is being checked
  ...SYSTEM WILL REBOOT AUTOMATICALLY

  The root file system has been corrupted or is inconsistent.
  - The system repairs itself. You do not need to intervene in any way.
  - Load the Mini System (...). Go to a shell from the main menu of the Mini System and check the root file system, e. g.:

    fsck -F ufs -l /dev/ios0/sdisk000s0

    If the root file system cannot be restored, or if data is lost in the process, you should restore the last system backup.

    If the root file system does not contain errors or has been restored, exit the Mini System via the following menu item:

    Abort Mini System and Reboot

- System startup is aborted directly after the start of the init process, and there is an immediate restart.

  The operating state has not been specified in the *initdefault* entry in the file */etc/inittab*.
  - Power the system up in single-user mode (...) and enter a valid operating state (generally **2** for multi-user mode):

    is:**2**:initdefault:

- After the message
  Checking channel_id configuration ...

  has been output, the RC procedures are halted, displaying the following message:
  Hit any key to go into single user mode

  The system has discovered an inconsistency between the channel ID areas in the NVRAM and those in the header of the system disk. These channel IDs are required for the logical addressing of the controller on the PCI bus and EISA bus.

  This discrepancy can be caused by replacing the motherboard, having defective NVRAM or using a different system disk.

  In some cases the system will be unable to continue booting even before this test is performed, reporting:
  PANIC: cannot mount root

  - To clear this problem you should proceed as described in Resolving inconsistencies in the NVRAM (RM200/300/400).

- System startup is aborted with one of the following messages:
  /usr sub-tree is not present□
  /var sub-tree is not present

  The */usr* or */var* file system is inconsistent and has therefore not been mounted.
  - Load the Mini System (...) and go to a shell from the main menu of the Mini System.
  - Check the inconsistent file system. The file system can be of type *vxfs* (following a reinstallation) or *ufs* (following an update installation), e. g.:

    fsck -F vxfs -l /dev/ios0/sdisk000s3□□□(for /usr)□
    fsck -F vxfs -l /dev/ios0/sdisk000s4□□□(for /var)□
    □
    fsck -F ufs -l /dev/ios0/sdisk000s3□□□□(for /usr)□
    fsck -F ufs -l /dev/ios0/sdisk000s4□□□□(for /var)

    If you specified the incorrect type, the system responds with invalid super block magic.

    If the root file system cannot be restored, or if data is lost in the process, you should restore the last system backup.

    If the file system does not contain errors and has been restored, exit the Mini System via the following menu item:

    Abort Mini System and Reboot

- Messages are displayed during the system start regarding manually checking a file system:
  /dev/ios0/rsdisk000s5: 0 BAD I=12547□
  /dev/ios0/rsdisk000s5: UNEXPECTED INCONSISTENCY, RUN fsck MANAULLY.□
  ufs mount: /dev/ios0/sdisk000s5: CONSISTENCY ERROR (This message contains the ufs or vxfs entry depending on the file system type.)□
  ufs mount: /dev/ios0/sdisk000s5 (Filesystem is dirty [run fsck]).□
  ...□
  Please check /home file system

  The reported file system is inconsistent or damaged. The system will be booted in single-user mode.
  - Enter the root password to get a shell:
    Type Ctrl-d to proceed with normal startup,□
    (or give root password for system maintenance):

    Check the file systems. You can use the *fsck*(1M) command to check all file systems contained in the */etc/vfstab* file. Clean, mounted file systems are skipped.

    The repaired file systems are still marked "dirty" after the first run through. You should consequently invoke the *fsck* command a second time.

    You can then boot the system in multi-user mode, using the *init 2* command (see *init*(1M)).

- System startup is aborted with the following message:
  S03swap: /etc/vfstab not found

  The */etc/vfstab* file does not exist.
  - Restore the file from the last system backup.

3.2.8 Outputting the HW_CONFIG defects table

The HW_CONFIG defects table is output on the console immediately before the message The system is ready. It contains information about faulty, defective and deactivated hardware components.

The defects table is also output on the screen each time somebody logs on as *root* (at Login or using *su - root*).

```
Disk: ios0/sdisk005(OS26(CD-ROM)): deactive set by system! Cause: set by LAR□
Disk: ios0/sdisk005(OS26(CD-ROM)): defect set by operator! Cause: deactivated□
Disk: ios0/sdisk005(CD-LC1(CD-RO): error set by system ! Cause: sdi 170 CMN□
Mobo: cab:0:Memory      :Slot:3: error set by system! Cause: cmn 34344: CMN
```

Each line contains the following information, separated by colons:

- Type of HW component.
- Device name or physical location. This location consists of a number of fields which are also separated by colons.
- Status of the component:

  error
  : The component is faulty, but is remaining in operation.

  defect
  : The component is defective and must be replaced.

  deactive
  : The component has been taken out of operation.

  The entry set by is followed by an indication of who reported the fault. If it was flagged manually this will be operator, otherwise system for the operating system, or the name of one of the SW components.
- Cause:

  The entry Cause: is followed by the reason for the entry. If this is given as an error code (such as *sdi 170*) you can use the command *getemm*(1M) to display a description of the encountered error, which will include instructions for resolving the problem.

We recommend you to carry out a further diagnosis using VConfig or Config (...), because this makes clear exactly where the HW component is installed and displays the full description of the error.

3.2.9 Login

3.2.9.1 Alpha terminals

The following problems can occur when logging in and working at alpha terminals. Notify Service if a suggested approach does not solve a problem.

- The login prompt appears again once the first character of the user ID has been input.

  The required Streams modules have not been configured.

  - As long as it is not the entry for the console that is involved, you can log in as the *root* user and execute the following command:

    autopush -f /etc/ap/*.ap (The **"*"** in the command line stands for the connection type.)

  - If it is the entry for the console, restore the file from the last system backup.

  The *autopush* entry is missing in the */etc/inittab* file. Different entries are required, depending on the type of connection:

```
ap::sysinit:/sbin/autopush -f /etc/ap/cons.ap□□□□□□□□# Console□
si::sysinit:/sbin/autopush -f /etc/ap/si.ap□□□□□□□□□□# SIM controller□
sih::sysinit:/sbin/autopush -f /etc/ap/sih.ap□□□□□□□□# SIH controller□
dptg::sysinit:/sbin/autopush -f /etc/ap/dptg2.ap□□□□□# DPTG2 protocol□
rtty::sysinit:/sbin/autopush -f /etc/ap/rtymux.ap□□□□# Remote TTY
```

  - As long as it is not the entry for the console that is involved, you can log in as the *root* user and correct

the */etc/inittab* file.

- If the entry for the console is missing, load the Mini System (...).

  Go to a shell from the main menu of the Mini System and mount the root file system:

  mount -F ufs /dev/ios0/sdisk000s0 /svr4

  The root file system is on */dev/ios0/sdisk000s0* by default. If your root file system is on another boot disk, you have to specify that.

  Correct the */svr4/etc/inittab* file.

  Exit the Mini System via the following menu item:

  Abort Mini System and Reboot

- The login prompt appears again when the user ID has been input.

  The */dev/tty* device file has been set up with the wrong major and minor numbers, or the device file does not exist.

  - Load the Mini System (...). Go to a shell from the main menu of the Mini System and mount the root file system, e. g.:

    mount -F ufs /dev/ios0/sdisk000s0 /svr4

    Delete the file */svr4/dev/tty* if necessary and recreate it:

    mknod /svr4/dev/tty c 130 0

    Exit the Mini System with the following option on the menu:

    Abort Mini System and Reboot

  The */etc/ap/*.ap* file is empty (The "*" in the command line stands for the connection type.).

  - Restore the file from the last system backup.

  The access permissions of the root file system have been changed with the result that the user no longer has read and execute rights.

  - Log in as *root* and check the access permissions:

    ls -ld /

    The access permissions should be set as follows:
    drwxr-xr-x 40 root root 1536 Apr 27 05:19 /

    Correct the access permissions if necessary with:

    chmod 755 /

- No prompt appears when the password has been input.

  If other users can still log in, the error probably lies in the *$HOME/.profile* file.

  - Log in as *root*, and correct the *.profile* of the user in question.

    If you cannot log in as *root*, you can bypass the *.profile* of the *root* user by logging in under another user name and issuing the *su root* command to get root permissions.

  If no user can log in, the error probably lies in the */etc/profile* file.

  - Load the Mini System (...). Go to a shell from the main menu of the Mini System and mount the root file system, e. g.:

    mount -F ufs /dev/ios0/sdisk000s0 /svr4

    Correct the */svr4/etc/profile* file.

    Exit the Mini System with the following option on the menu:

    Abort Mini System and Reboot

- The login prompt and inputs are incomplete or garbled.

  The */etc/ap/cons.ap* file does not exist or has been destroyed.

  - Restore the file from the last system backup.

- Characters that have been keyed in do not appear on the screen.

  Screen output has been stopped.

  - Enable screen output again by pressing [CTRL][Q].

- You see the following message, and then the login prompt appears again.
  Unable to change directory to "<*directory*>"

  The directory assigned to the user ID in the */etc/passwd* file does not exist or has incorrect access permissions.
  - Log in as *root*, check the entry in the */etc/passwd* file, and the access permissions for the directory, and correct them as appropriate (see *chmod*(1), *chown*(1), *chgrp*(1)).

  The directory assigned to the user ID in the */etc/passwd* file has not been mounted.
  - Log in as *root* and execute the *fsck*(1M) and *mountall*(1M) commands (both without any options).

  If you cannot log in as *root*, you can bypass the *.profile* of the *root* user by logging in under another user name and issuing the *su root* command to get root permissions.
- You see the following message, and then the login prompt appears again:
  No Shell

  The executable program assigned to the user ID in the */etc/passwd* file cannot be loaded.
  - Log in as *root*, check the ownership and access permissions of the program and the associated path components, and correct them as appropriate.

  The directory assigned to the user ID in the */etc/passwd* file has not been mounted.
  - Log in as *root* and execute the *fsck* and *mountall* commands (both without any options).

  If you cannot log in as *root*, you can bypass the *.profile* of the *root* user by logging in under another user name and issuing the *su root* command to get root permissions.

- The following message appears when the user ID and password have been input:
  Login incorrect

  An incorrect user ID and/or password has/have been input.
  - When the login prompt reappears, enter a valid user ID with its associated password.

    The actual keyboard allocation may not match the labeling on the keyboard because of the way your keyboard has been configured. In this case, some characters (special characters in particular) will have been moved to other keys.

    If you cannot remember the password, log in as *root* and delete the password:

    passwd -d -f <*user_id*>

    At the next login, the user is prompted to specify a new password.
  - If you cannot remember the password of the *root* user, load the Mini System (...). Go to a shell from the main menu of the Mini System and mount the root file system, e. g.:

    mount -F ufs /dev/ios0/sdisk000s0 /svr4

    Delete the password from the */svr4/etc/shadow* file.

    Exit the Mini System with the following option on the menu:

    Abort Mini System and Reboot

    In order to protect your data, you should make sure that no unauthorized persons can work with a Mini System on your machine.

  The */var/adm/lastlog* file is inconsistent.
  - Power the system up again.

- The login attempt is denied with the following message:
  rlogind: cannot create utmp entry

  There is no more free space in the */var* file system so no more log entries can be written to the */var/adm/utmpx* file.
  - Clean up the */var* file system by deleting superfluous files (e. g. in */var/tmp*). You may have to set up a larger partition for */var* (see the "System Administrator's Guide").

    If you are not logged in as *root* and no other users are logged in so you cannot get root permissions with *su root*, you have to load the Mini System (...). Go to a shell from the main menu of the Mini System, mount the root file system, e. g.:

    mount -F ufs /dev/ios0/sdisk000s0 /svr4

    Clean up the file system */svr4/var*.

    Exit the Mini System with the following option on the menu:

    Abort Mini System and Reboot

- The login process takes an unusually long time.

  The */var* file system is possibly very full.
  - Use the *df -k* command (see *df*(1)) to check the utilization of the */var* file system.
  - If the file system is very full, clean it up as described in the item above.

3.2.9.2 Graphics terminals

The following problems can occur when logging in and working at graphics terminals. Notify Service if a suggested approach does not solve a problem (Please refer to the "SINIX/windows User Environment" manuals for detailed information about working with graphics terminals.).

- The screen is gray (possibly with the X logo) or black or shows other images.

  The screen saver is running.
  - Press any key or a mouse button.

- The screen goes black at the end of the system startup process.

  The wrong type of graphics controller has been set up.
  - Go into alpha mode (...).

    Any messages about the error are output in alpha mode.

    Check the settings in the SYSADM user interface under *console_config — Hardware*, and change them as appropriate.

  The graphics controller is defective.
  - Notify Service.

- The system will not start a graphics monitor. Nor is it possible to switch to graphical mode. No error message is output, though.

  The X Window package has not been installed. The terminal type *VGA-Graphic* may not have been selected when the operating system was installed.
  - Install the required software packages (see the "Reliant UNIX Operation" manual).

  The start option *xdm* has not been enabled.
  - Enable the start option in the SYSADM user interface under *console_config — xserver — Startup*.

  The start procedures are not executed.
  - Check the *rc* procedures *S99motif*, *S72admcon* and *S82mouse*.

Processes are hanging.

- Check the processes with the *ps -ef* command (see *ps*(1)). This ought to display an *xdm* process, and an X server process and another *xdm* process as child processes. If applicable, delete the processes with the *kill -9* command (see *kill*(1)), and restart the *xdm* daemon and the X server with *sh /etc/rc2.d/S99motif start*.

Information about your computer is missing from the file */etc/hosts*.

- Check this file and add to it if necessary.

  For computers which are operated in a LAN, the file */etc/hosts* must contain an entry *localhost* with the name of your own machine and its correct Internet address:
  127.0.0.1 localhost
  139.4.123.7 monaco

  For computers with no connection to a LAN the address of the local host is used instead of an individual Internet address:
  127.0.0.1 localhost
  127.0.0.1 monaco

- The following message appears when the user ID and password have been input:
  Login incorrect

  An incorrect user ID and/or password has/have been input.

  - When the login prompt reappears, enter a valid user ID with its associated password. If you cannot remember the password, log in as *root* and delete the password:

    passwd -d -f <*user_id*>

    At the next login, the user is prompted to specify a new password.

- If you cannot remember the password of the *root* user, load the Mini System (...). Go to a shell from the main menu of the Mini System and mount the root file system, e. g.:

  mount -F ufs /dev/ios0/sdisk000s0 /svr4

  Delete the password from the */svr4/etc/shadow* file.

  Exit the Mini System via the following menu item:

  Abort Mini System and Reboot

  
  In order to protect your data, you should make sure that no unauthorized persons can work with a Mini System on your machine.

The keyboard configuration does not match the keyboard that is connected.

- Check the keyboard settings in the SYSADM user interface via *console_config — keyboard — set*, and change them as appropriate.

- When the user ID and password have been entered, first the screen goes gray and then the login window appears again.

  The *$HOME/.xsession* file is incorrect.

  - Log in as *root* and rename the user's *.xsession* file. This causes the default file */usr/lib/X11/xdm/Xsession* to be called automatically after login.

  The user's home directory is invalid.

  - Check that the home directory exists and, if so, check its access permissions.

  The installation is not right. An error has occurred in the *rc* procedures.

  - Check the installation. (see the "Reliant UNIX Installation" manual and the SINIX/windows documentation)

- The mouse manager outputs the following message:
  mousemgr: cannot open /dev/m320.□
  Please check if all cables are connected properly,□
  Xserver start failed

  An incorrect mouse driver has been configured.
  - Check the configuration (see chapter "Installing the system unit and console" in the operating manual for your computer).

  The mouse manager does not start properly.
  - Check whether the */etc/rc2.d/S82mouse* file exists and, if necessary, restore it from the last system backup. If the file does exist, try to start the mouse manager by executing the following commands in this order:

    cd /etc/rc2.d□
    sh S99motif stop□
    sh S82mouse stop□
    sh S82mouse start□
    sh S99motif start

  The mouse is defective.
  - Replace the mouse.

- The window or font sizes or colors are not those that are wanted.

  Resource definitions have been changed.
  - Change the definitions as required (see the SINIX/windows documentation).

Protocol files

You can find more information on error situations in the following log files:

$HOME/.xerrors
: *xdm* messages about the current session

$HOME/.xerrors.old
: *xdm* messages about the previous session

/usr/lib/X11/xdm/xdm-errors
: System-wide error file for *xdm*

/usr/lib/X11/xdm/xdm-pid
: File with the process ID of the current *xdm*

/var/sadm/install/logs/.log*
: Installation error files

/usr/adm/X<n>.msg
: X server error log file (*n* = display number)

$HOME/.dt/startlog
: Default session log (TED)

$HOME/.dt/errorlog
: *xdm* messages for TED session

Possible error messages in the log files:

- can't open display

  The *DISPLAY* variable has not been set correctly.
  - Go into alpha mode (...) and set the variable. If you are using an X terminal or an Rlogin connection, you have to specify the name of the X terminal or local computer:

    DISPLAY=<*name*>:0.0 export DISPLAY

  The *xhosts* command was not executed for an Rlogin connection.
  - Go into alpha mode and execute the following command on the local computer (specify the name of the

remote computer as *name*):

xhosts +<*name*>

- cannot convert internet address ... to host name

  No computer name has been assigned to the Internet address.

  - Check the Internet address in the */etc/hosts* and *$HOME/.rhosts* files, and correct it as required.

- can't lock pid-file, another xdm is running

  An old *xdm* process is running which cannot be stopped with *S99motif stop*.

  - Get the process ID of the old process, terminate it, and restart the GUI:

    cat /usr/lib/X11/xdm/xdm-pid
    kill -9 <*process_id*>
    sh /etc/rc2.d/S99motif start

- toolkit II error

  The node name of the system has not been entered correctly in the */etc/hosts* file, or has not been entered at all.

  - Check the entry in the */etc/hosts*, and change it as appropriate.

- broken pipe

  The pipe between the X server and X client has ben broken by the system or by an application. This message occurs, for example, if you exit a window with *xkill*.

  - To avoid this message, you should exit the X client in the defined way (see the SINIX/windows documentation for the respective client).

3.3 System running

3.3.1 System environment monitoring

The computer and the peripherals cabinets possess sensors for detecting critical temperatures, primary power failure, defects in fans and power packs etc.. When a fault occurs, the environmental monitoring hardware generates an interrupt. The EEC daemon (enhanced enclosure control) collects information from the environmental monitoring hardware, generates error reports, passes information to diagnostic programs, and presents add-on software with a uniform interface to the various monitoring controllers.

In the E models, the controller for environmental monitoring is called i960 (for the system cabinet) and SAF-TE for the peripherals cabinet. Earlier models have an EIP or ICP. The operating system addresses these controllers via the EEC daemon.

E models may also have IDC and EIP3 controllers installed, but these are required only for switchover functionality in high-availability configurations and not for environmental monitoring.

If you have the package *SIpups* installed, the UPS will be monitored by the PUPS daemon (see *pupsd*(8)). This daemon controls the behavior of the UPS according to the specifications in the file */var/adm/ubft/_upstab*.

Changes to the file */var/adm/ubft/_upstab* are put into effect by passing signal □1 to the EEC daemon : *kill -1 <process_number>*.

Any fault which occurs is

- Reported on the console (e.g.):
  eecd[471]: eecd:10001: CAB15: FAN FAILURE. Shutdown in 10 seconds.
- Dealt with by the EEC daemon as specified in the file *_upstab*(4), which you can modify as appropriate for your system.

  If, for any reason, the EEC daemon is not running, the system or the relevant cabinet will respond to total failure of its fans by closing down automatically after a period of 5 minutes. Since this is not a proper shutdown it may cause data to be lost. In such a case you should power the system down yourself.
- Reported to Logging V3.0 (...).
- Reported to ServerView (...).
- Displayed under VConfig and Config (...).
- Indicated on the LCD and/or LED of the relevant cabinet:
  - The LED *ON/BATT* on the system cabinet (Only applies to the model RM400-xxx (Tower).) or the LEDs *ON/BATT* and *FAIL* on a peripherals cabinet light up red (only applies to cabinets with an internal BBU).

    The internal BBU is active in response to a mains power failure or a fault in a power supply unit.
  - The LED *FAIL* on a cabinet lights up red .

    The room temperature is too high, or the cabinet's ventilation ducts are obstructed, or there has been a total failure of the fans.

  ▶ Remedy the malfunction, or notify Service so that defective components can be replaced.

Messages from the EEC daemon - the general form

All the messages issued by the EEC daemon re laid out in the following way:

stamp: eecd[*pid*]: eecd:*xxxxx*: CAB*yy*: *description* (*aa*, *bb*) *zzz*

- *stamp*

  Time stamp and system information from the logging service

- *pid*

  Process ID of the EEC daemon

- *xxxx*

  10001 = Fault occurred
  10002 = Fault cleared
  10005 = Warning
  10008 = Warning withdrawn

- *yy*

  Physical ID of the system cabinet or peripherals cabinet

- *description*

  Message text (see the following list)

- *aa, bb*

  Additional information, depending on the specific fault occurrence.

- *zzz*

  is fixed
  : The fault has been cleared

  Shutdown in *nnn* seconds
  : This text is output if a shutdown is specified for this fault occurrence in the file *_upstab*

  Shutdown has been cancelled
  : The fault was cleared during the countdown to shutdown; the shutdown will not be carried out.

Messages from the EEC daemon - causes and corrections

- POWER FAILURE

  The primary power supply has failed.

  - Check your mains power lines and fuses.

- TOO FEW POWER SUPPLIES

  One of the redundant power units has failed.

  - Have Service replace the defective power unit.

- POWER SUPPLY FAILURE

  The (remaining) power unit is faulty.

  - Have Service replace the defective power unit.

- POWER FAILURE - UPS ACTIVE

  The primary power supply has failed; the UPS or BBU is active.

  - Check your mains power lines and fuses.

- Battery low

  The rechargeable battery in the UPS or BBU is almost discharged.

  - The system will perform a shutdown itself, so that the operation is completed before the UPS or BBU is completely discharged.

- FAN WARNING

  One fan is defective; the remaining fans are maintaining ventilation.

  - The system can continue in operation, but you should have Service replace the defective fan in any case.

- FAN FAILURE

  One fan has failed; adequate ventilation is no longer guaranteed.

  - Check whether the fan will start up again, by closing down the system unit and then switching it on again. You should have Service replace the defective fan in any case.

- WARNING TEMPERATURE

  The temperature is running high, but the system can remain in operation.

  - Check whether the ventilation slits are obstructed and whether the current room temperature is within the permitted range.

- CRITICAL TEMPERATURE

  The temperature is too high; the system will be turned off.

  - Check whether the ventilation slits are obstructed and whether the current room temperature is within the permitted range. When the cause of the fault has been dealt with you can switch the system on again.

- DOOR STATUS

  Intruder alert: the door is being opened or closed.

  - You can reset this alert by issuing the following command:

    /usr/bin/lar/del_hw_config -c -A -a*value*

    The value that is entered after *-a* depends on the configuration. You can establish it by entering:

    /usr/bin/lar/del_hw_config -C

- NEW CABINET FOUND

  The system has encountered a system cabinet or peripherals cabinets which was not previously in the system configuration.

  If you have not modified the configuration, then the address of an existing cabinet has been altered.

  - If necessary, restore the previous situation.

- CABINET NOT ON

  One of the cabinets which is recorded in the system configuration is not switched on or is no longer present.

  - Switch the cabinet on, or reconnect it.

- DOUBLE CABINET ID

  The system has encountered a second cabinet with the given ID. This second cabinet will not be monitored.

  - Give the new cabinet one of the available IDs.

- LOST CONNECTION WITH ONE CONTROLLER

  One of the controllers for the environmental monitors is no longer responding. You will see this message if one of the cabinets is switched off, or the connection to a cabinet is interrupted, while the system is in operation.

  - Switch the cabinet on, or reconnect it.

 You can use VConfig or Config to determine which cabinet contains a faulty component (...).

Messages from the PUPS daemon

Events which affect the uninterruptible power supply (UPS) are output in the following way:

pupsd[10527]: pupsd10005: *message*

The following list gives you an overview of the possible messages and their significance:

- BYPASS ACTIVE

  The UPS is being operated in bypass mode; this does not give the system any protection from power failures. Please refer to the operating instructions for your UPS to see how to deactivate bypass mode.
- BYPASS ACTIVE is fixed

  Bypass mode has been turned off; the UPS is again protecting the system from power failures.
- POWER FAILURE-UPS ACTIVE Shutdown in *xxx* seconds

  The primary power supply has failed; the system will be shut down after the stated number of seconds.
- POWER FAILURE-UPS ACTIVE is fixed. Shutdown has been cancelled

  The primary power supply has been restored; the system will not be shut down.
- <PWR>: WARNING: Battery low!

  The battery voltage is too low.
- <PWR>: System waits *xxx* minutes until battery is loaded

  The system will wait the specified number of minutes, until the battery is sufficiently charged, before continuing with the boot sequence.

3.3.2 Reliant UNIX error messages

3.3.2.1 Error handling

Reliant UNIX errors are handled by the system as follows.

- System errors, unrecoverable SCSI device errors and application errors which do not cause a system halt are written to the STREAMS device */dev/log* by error reporting routines (e. g. *cmn_err()*, *drv_err()* or *syslog*(3C)). The *syslogd*(1M) daemon reads */dev/log*.

  Applications which incorporate the Logging V3.0 application programming interface report errors directly to Logging V3.0 without using the syslog daemon (see section Passing data directly to Logging).

  The syslog daemon handles these errors in accordance with parameters which are defined in the *syslog.conf*(4) file and which you can adjust to your requirements. Errors are reported on the console, written to various log files (e. g. */var/adm/log/messages*) and reported to Logging V3.0, depending on the entries in the *syslog.conf* file.

- Recoverable SCSI device errors are written to the STREAMS device */dev/error* which is read by the *errord*(8) daemon. The error daemon stores these messages in the */var/adm/error/errfile.current.Z file*. Older files are given the name *errfile.previous.Z* and later *errfile.<date>.Z*. A user interface is at your disposal to evaluate these messages (...).

  In addition, the errors read by the error daemon are passed to the *errlogd* daemon which monitors threshold limits for SCSI devices. The error daemon distinguishes between two types of threshold limits:
  - Soft limit: The device is errored but remains in operation
  - Hard limit: The device is defective and has to be replaced

  When one of these threshold limits is reached, the error is output on the console and reported to Logging V3.0.

- All error messages reported to Logging V3.0 are collected in various log files and may be passed on to Event Management, DSM Event Reporting or TransView.

  In response to error messages, Event Management can start reaction programs in which the handling of the error message is defined (...).

  DSM Event Reporting is a separate program package for error diagnostics in SNA networks (...).

  TransView is a separate program package for system and network administration, and administration of applications in complex networks (...).

3.3.2.2 Structure of error messages

Reliant UNIX errors are output on the console with the following syntax:
[*reporting component*:] [*error weight*:] [*error identifier*] *error text*

Example:
CMN: NOTICE: sduart 50 SCC C50 duwx16552_devlock: serial port is busy (minor=5), (IRQ4) = 0x80311c20.

The reporting component is only displayed if the error is reported via the *cmn_err()* (CMN = kernel) or *drv_err()* (DRV = driver) error reporting routine.

The error weight is also only displayed if the error is reported via the *cmn_err()* or *drv_err()* error reporting routine:

Kernel	Driver	Display when getemm is called	Meaning
PANIC	Fatal	EMERG	System halt
-	Err1	ALERT	Error indicating considerable loss of data and requiring immediate action
-	Err2	CRIT	Error indicating loss of data or malfunctions

			requiring action soon
-	Err3	ERR	Error indicating malfunctions but not requiring immediate action
WARNING	Err4	WARNING	Warning that normal functioning of the system could be impaired unless the appropriate action is taken
NOTICE	Warn	NOTICE	Indication of a possible problem
-	Note	INFO	Status message
-	Dbg1-Dbg5	DEBUG	Messages only used in program debugging

If an error identifier is output, you can use the *getemm*(1M) command to see a description of the problem and handling instructions to solve it. An error identifier always consists of the name of the reporting module in the component□
(*sduart* in the example above) and a unique error number relating to the module (*50* in the example above).

Example of the *getemm* call and its output:

getemm sduart:50□
Text:□
SCC C50 duwx16552_devlock: serial port is busy (minor=%d), (IRQ%d) = 0x%x□
Description:□
[sduart:50: SCC C50 duwx16552_devlock]□
It is not possible for the sduart driver to set the interrupt□
routine for the 16552 serial controller into the isa/eisa□
interrupt table. The 16552 serial controller only has one□
unique interrupt number (channel 0 = intr#4,□
channel 1 = intr#3).□
"minor" indicates the sduart channel trying to set an interrupt□
(5=channel 0, 6=channel 1).□
"IRQ"indicates the requested interrupt number.□
Action:□
Some other driver uses the requested interrupt. Check if the□
UPS driver is configured and active.□
The user must decide which driver should use the serial port:□
- When UPS is active the user must disable some "LOGIN"s at□
this port□
- If UPS was active and the user wishes to use the serial port□
for a terminal again, he must remove the UPS package before.□
Weight: WARNINIG

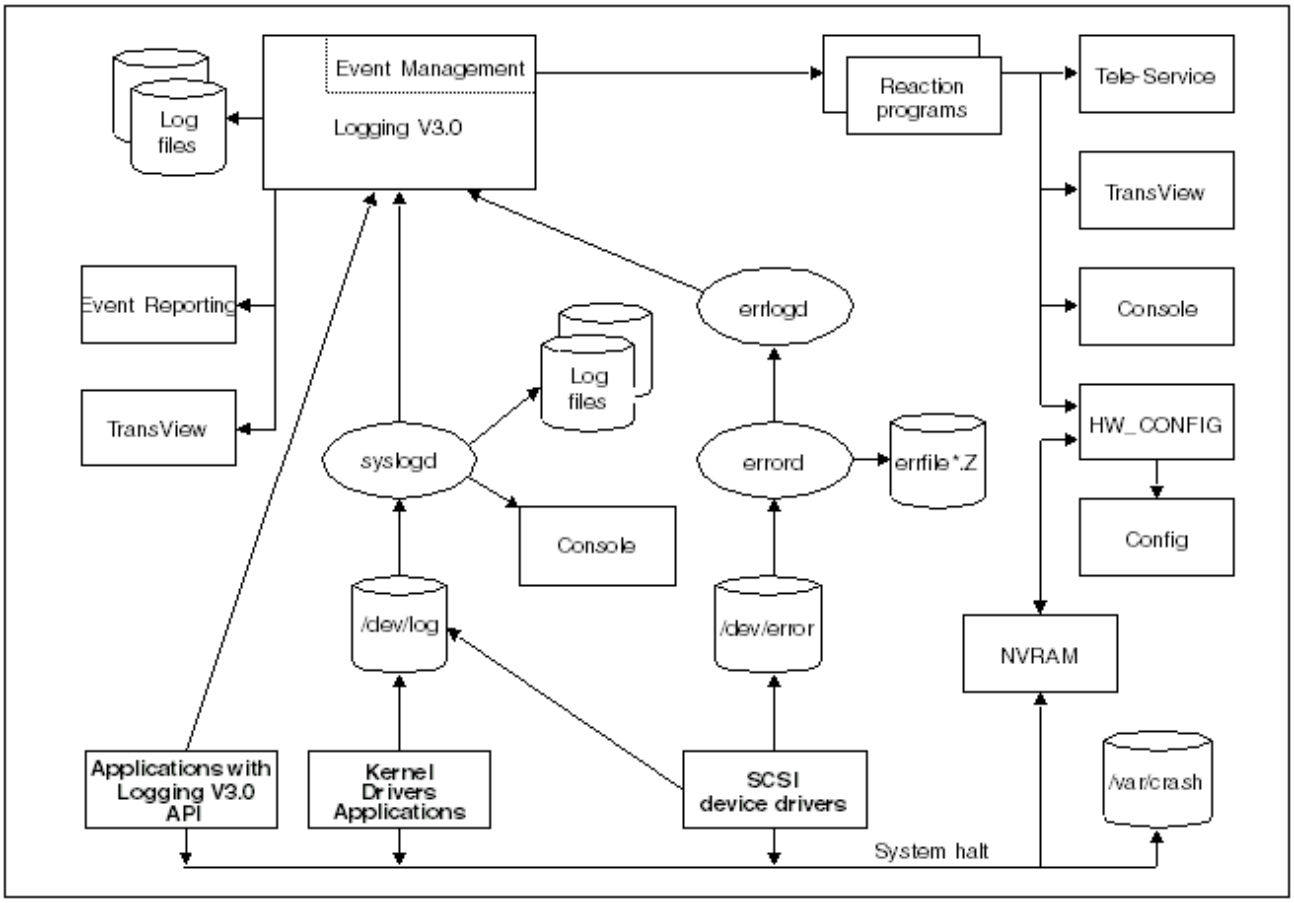


Figure 2: Data flow of error messages

3.3.3 System halts

If an error causes a system halt, this can be a serious software or hardware error, or a result of irregularities in operating system software that can be remedied by restarting the system. A system halt is primarily a protective function to avoid loss of data.

What happens in the event of a system halt:

- The *ikdb*(8) debugger is started:
  You are now in the kernel debugger. Type ex to exit
  ikdb>
  - Enter the *ex* command.
    The following message is displayed:
    Type ndp for remote debugging
  - Enter the *ex* command again. The system is then automatically powered down and then booted again.

What happens during shutdown:

- The system buffer is written back (see *sync*(1M)).
- A core dump is written to the swap area.

What happens during power-up:

- The core dump (*dump.<x>*), the file (*unix.<x>*) containing the executable system kernel and an archive (*info.<x>*) (The "x" in the *dump*, *unix* and *info* file names stands for a serial number.) containing a series of compressed system files are written to the standard */var/crash* directory. This directory is defined by the *DUMPDIR* variable in the */etc/default/dumpsave*(4) file. You can modify it (see *dumpsave*(1M)).
- The console log that was saved in the NVRAM is written to Logging V3.0.

Please save the *dump.<x>* and *unix.<x>* files and the *info.<x>* archive, for instance to a tape cartridge, and forward this medium to Service. You should also include a brief description of the circumstances in which the system halted. If a kernel other than *unix* (e. g. *unix.old*) was active at the time of the system halt, then this kernel must be included in the error documentation because the core dump cannot be evaluated otherwise. Make sure that there is always enough space in the */var* directory for a new core dump to be stored there.

3.3.4 System hangups

Hardware and software errors can put Reliant UNIX in an inoperable state which will not return to normal even after a longer wait. Before taking "radical" action, though, you should check whether the system as a whole has come to a standstill, whether just some workstations (terminals, PCs) are blocked, or whether an application is hanging.

- Try to log in at another screen as the *root* user. If you manage to do this, first save all data that is in memory to disk using the *sync*(1M) command. Then try to shut down the system properly.
- If you cannot log in as the system administrator or if the system is blocked, try to start the *ikdb*(8) debugger on the console:
  - On an RM300/400 briefly press the RESET button while you are in the IKDB. The keyswitch must be in the LOCAL position.
  - On an RM200 you have to go into alpha mode:
    - Press and hold down the [ALT] key, press the [SysRq] key (international keyboard) or the [S-Abf] key (German keyboard), release both keys and then press the [H] key.

    Then enter the following key combination:
    - Press and hold down the CTRL key and type "@". Release the [CTRL] key and enter the characters **IKDB** (in upper case) directly afterwards (no blanks).

  Terminate all application processes that are running and exit the debugger:

```
ikdb> ki
ikdb> ex
```

  If the debugger cannot be exited, enter the *pa* command. If the system is still blocked after exiting the debugger, call the debugger again, enter the *pa* command and exit the debugger:

```
ikdb> pa
ikdb> ex
```

  Exiting the debugger now triggers a system halt and control returns to the debugger. Exit the debugger again:

```
...
Type ndp for remote debugging
ikdb> ex
```

  The following happens if one of the two actions (terminating all running processes or triggering a system halt) is successful:
  - The system buffer is written back (see *sync*(1M))
  - A core dump is written to the swap area
  - The system is automatically restarted. During the restart, the core dump (*dump.<x>*), the file (*unix.<x>*) containing the executable system kernel and an archive (*info.<x>*) (The "x" in the *dump*, *unix* and *info* file names stands for a serial number.) containing a series of compressed system files are written to the standard */var/crash* directory. This directory is defined in the *DUMPDIR* variable in the file */etc/default/dumpsave*(4). You can modify it (see *dumpsave*(1M)).

Please save the *dump.<x>* and *unix.<x>* files and the *info.<x>* archive, for instance to a tape cartridge, and forward this medium to Service. You should also include a brief description of the circumstances in which the system halted.

If a kernel other than *unix* (e. g. *unix.old*) was active at the time of the system halt, then this kernel must be included in the error documentation because the core dump cannot be evaluated otherwise.

Make sure that there is always enough space in the */var* directory for a new core dump to be stored there.

- If the measures described above are not successful in resolving the problem, you may have a serious hardware or software error. In this case, please contact Service.

  On an RM300/400, press the RESET button for more than 3 seconds and power up the system again. If there is no response you have to turn the system off and back on again.

  As the RM200 does not have a RESET button the only thing you can do is turn the system off and back on again.

  With these actions, however, you must reckon on data inconsistencies after the restart.

3.3.5 System bottlenecks

System bottlenecks are indicated by delayed responses to screen inputs, applications taking a long time to respond and, in some cases, warnings on the console.

There can be many reasons for such bottlenecks or drops in performance, some of which are given in the list below.

- File systems are almost full
- Swap space is exhausted
- Memory capacity is exhausted
- Long processes hogging the CPU
- Space in core tables is exhausted
- Too many simultaneous system users
- Performance problems on the application level
- Poor program flow (loops, locks etc.)

Some commands and desktop menus are available in such situations to pinpoint the causes and prepare for remedies:

- Check the utilization of the file systems with the *df*(1) command.
- Check the swap space with the *swap*(1M) command.
- Check accesses to disk with the *dkstat*(1M) command.
- Use the *sar*(1) command to display a sample of CPU utilization, e. g. **sar 6 10** (10 measurements each 6 seconds apart).
- Get an overview of users who are logged in(1) and their processes with the *who*(1), *whodo*(1M) or *finger*(1) commands.
- Use the *ps -ef* command (see *ps*(1)) to see a detailed process list, and scan this for processes that have been running a long time (high figures in the TIME column).

Bottlenecks may arise periodically at certain times. In such cases, it is advisable to observe the situation over a longer period of time. Find out whether any conversions (software or hardware) or changes to the configuration of your system have been carried out since the bottlenecks occurred.

Generally your system should have been configured optimally for its intended usage. Extended usage cannot always be foreseen, however, and can take the system to its performance limits. In this case, it is also necessary to consider expanding capacity or increasing performance, e. g. adding memory or hard disks or installing a more powerful model.

Refer to the "Tuning Guide" for further information in this regard. Our specialists will be pleased to help you with special measurement tools to provide performance analyses and possibly tune the system. Contact Service and have them handle these activities.

4 Evaluation and administration of logbooks

The Logging system collects information on the progress, status and any error situations of system and application programs in logbooks.

This chapter gives system administrators and Service employees who are involved in system diagnostics an overview of work with Logging under the *WebSysAdmin* user interface.

Using this user interface, which is available both on a local computer and in an administration domain (*DomainAdmin* software product) you can evaluate and process log entries (*Event Management*) and administer the Logging system itself.

The Section "System level" contains information for programmers who want to use Logging in their own application programs or want to extend the properties of Logging.

4.1 User interface

This chapter describes the Logging system under the *WebSysAdmin* user interface. You can call system diagnostics with Logging via *Application Diagnostic* from the menu bar. This user interface provides you with extensive help texts.

4.1.1 Starting the user interface

WebSysAdmin can be called both in write mode and in read mode. Note that *WebSysAdmin* can only be called in write mode on a server by one user at any one time. If you want to start other *WebSysAdmin* users on the same server, this is only possible in read mode.

WebSysAdmin needs a minimum screen resolution of 1024 x 768 to be able to display all information in a readable format.

4.1.1.1 Starting on a PC under Windows 95/98/NT

- Launch your browser.
- Enter the following in the address bar in order to call the program in write mode:

  http://*Computer_name*:8881

  Enter the following in the address bar in order to call the program in read mode:

  http://*Computer_name*:8882

  The user interface is loaded by the specified computer and the welcome screen appears on your monitor.
- Select the dialog language that you want (default = English) and then click the *Diagnostic* button at the bottom left. A dialog box appears in which you have to enter the root user id (read mode only) and the root password of the chosen computer.

4.1.1.2 Starting on the LAN console

- You start the user interface in write mode with the following command:

  # /opt/wsa/bin/wsa *Computer_name*
- You start the user interface in read mode with the following command:

  # /opt/wsa/bin/wsa –r *Computer_name*

Then the welcome screen appears and you continue as described above.

4.1.2 Overview of functions

When you have started the Logging system (...) any errors that occur with the individual components are reported to Logging in the shape of log entries (...) and stored there in logbooks (...). The log entries are assigned to the individual logbooks via component assignments (...). You can view various items of status information to check whether Logging is running correctly (...).

Various filter options (...) are available so that you can evaluate log entries.

Log entries can be passed to Event Management (...). Events for any errors that may occur are defined in the Event Management facility (...) and reactions to these events are triggered (...). You can administer the definitions of events and reactions. You can also query and initialize counters for the individual events (...).

Log entries can not only be passed to Event Management but also be stored in an ASCII file for passing to an administration tool such as TransView or Unicenter TNG (...).

The monitoring facility monitors thresholds for SCSI devices, memory boards and caches on CPU under certain conditions (...). It checks whether the frequency of errors has reached a critical level which is no longer acceptable as far as data integrity is concerned.

4.1.3 Logging server

4.1.3.1 Starting and stopping

The logging server is started automatically when the system is booted, and is stopped when it shuts down. However you can stop the logging server and restart it from the *Log Administration* tab which is called via the item of the same name on the menu bar, for instance because you want to change the start options.

Figure 3: Menu for starting/stopping the logging server

If you are in an administration domain then these two actions can be carried out both for the entire administration domain and for each node. This depends on your selection in the user interface's hierarchy browser.

Logging can only be active once on a system at any one time. If you restart the logging server although it is already running, a restart is performed automatically. Information on the start and stop times is recorded in the *log3debug* logbook.

Any incorrect behavior by Logging itself is stored in *debug_log3* which is located in the base directory (*$LOGPATH;* see Figure "Dialog box for displaying the status of Logging").

4.1.3.2 Changing start options

When you select *Start of logging server* you can change the settings for the start options.

The first two options have an impact on the further processing of log entries: when one of these start options has been enabled and the component assignment flag has been set (...), the log entries are passed to Event Management or stored in an ASCII file for processing by an administration tool.

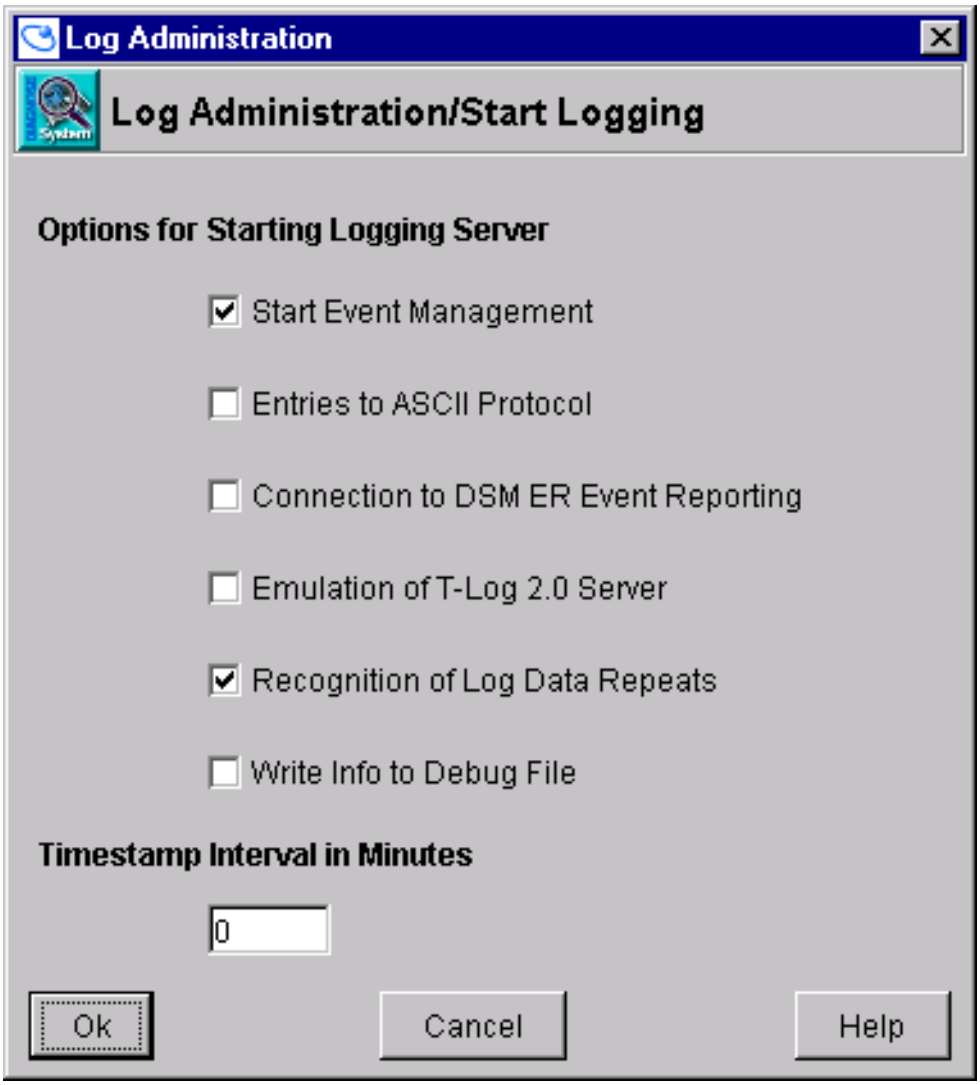


Figure 4: Dialog box for setting start options

4.1.3.3 Displaying the status

You can display various items of information on the status of Logging by opening the *Log Administration* tab followed by the *Status* subtab. In an administration domain the information is displayed for every node.

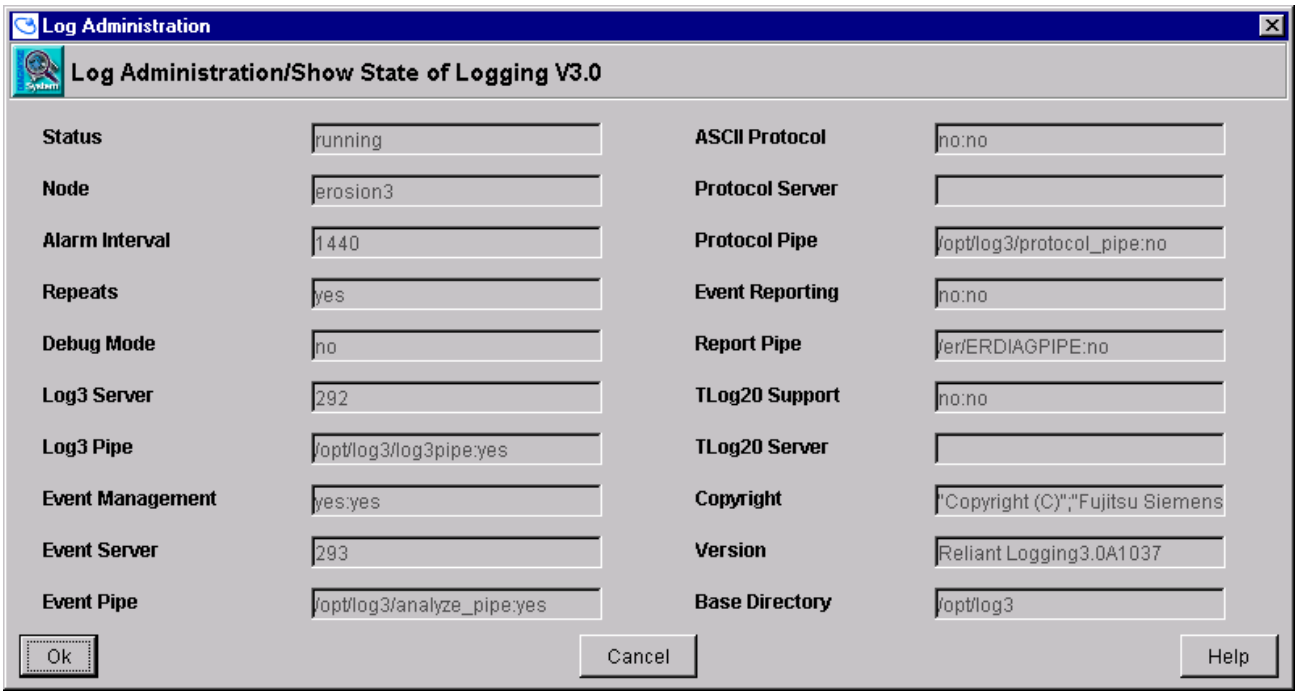


Figure 5: Dialog box for displaying the status of Logging

In addition to the current status of Logging you see, among other things, which start options have been set, which services have been started, which pipes exist and what they are called, which version of Logging has been installed and where the base directory is (*$LOGPATH*).

4.1.4 Components and logbooks

4.1.4.1 Administering component assignments

Components consist of one or more modules. Modules can be program packages, individual applications or parts of the system software, e.g. the software for individual controllers or drivers. For example, the entire driver software is assigned to a component, whereas the software for every driver constitutes a module. If a component consists of just one module, then the terms are used synonymously.

Fujitsu Siemens assigns an internationally unique number ranging from *1* to *4294967295* to every component. These numbers are used in assigning components to logbooks. If you want to assign numbers to components of your own, please get in touch with the Operating Systems Development department at Fujitsu Siemens (stating *Logging* as the subject) to get numbers that have not been assigned yet.

You administer component assignments by opening the *Log administration* tab followed by the *Components Assignments* subtab.

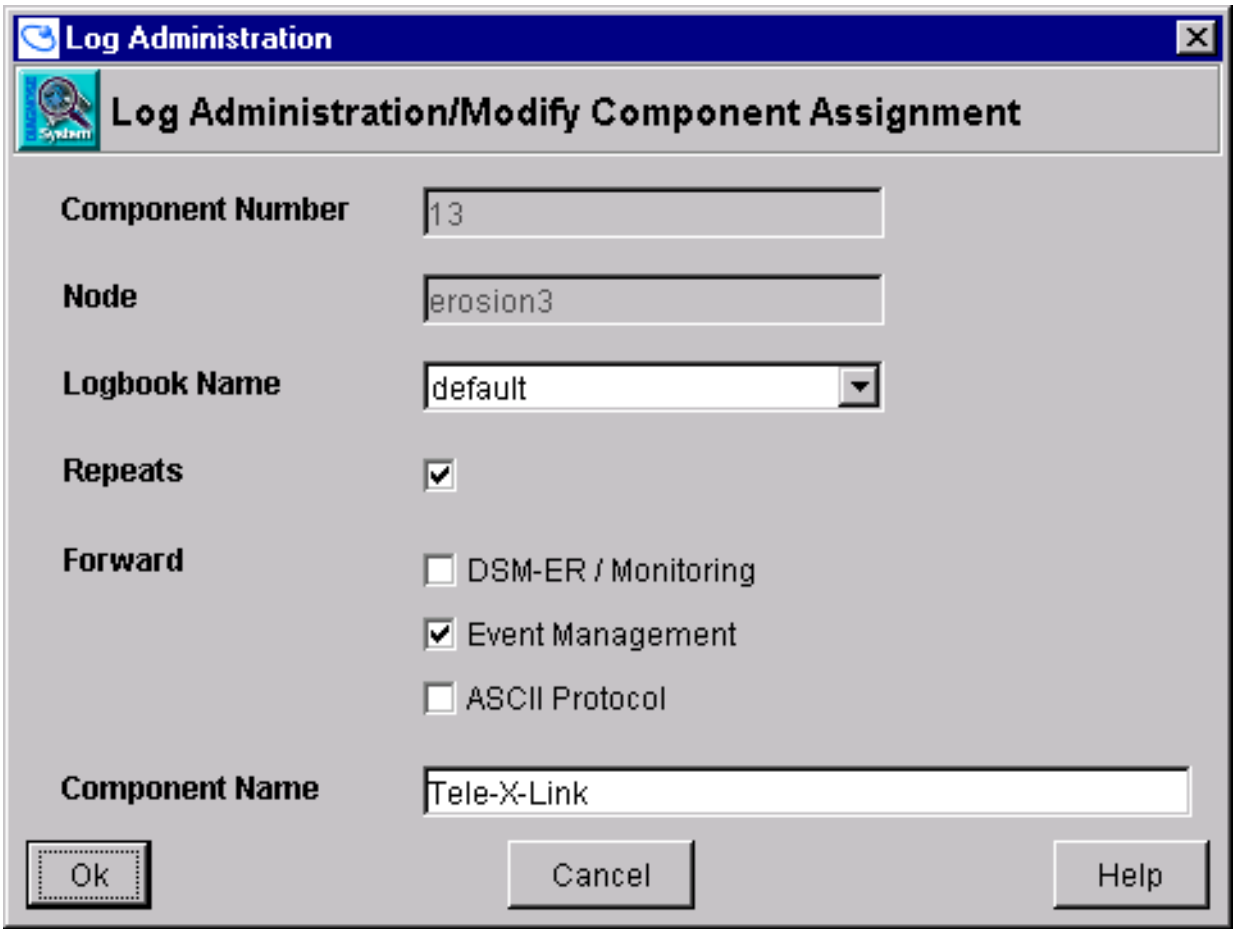


Figure 6: Dialog box for modifying a component assignment

The assignment of components defines the logbooks to which individual log entries are written, and whether log entries for a component are to be passed on. They are only passed on, though, if the corresponding option has been enabled when the logging server starts, e.g. log entries can only be passed to Event Management if that facility is running.

If you delete the assignment of a component to a logbook the log entries for this component are not lost. They are written to the *default* logbook.

4.1.4.2 Administering logbook definitions

All information on the progress, status and any error situations of system and application programs is collected in logbooks which consist of a cycle of log files.

You administer logbook definitions by opening the *Log Administration* tab followed by the *Logbooks* or *Default Logbook* subtab.

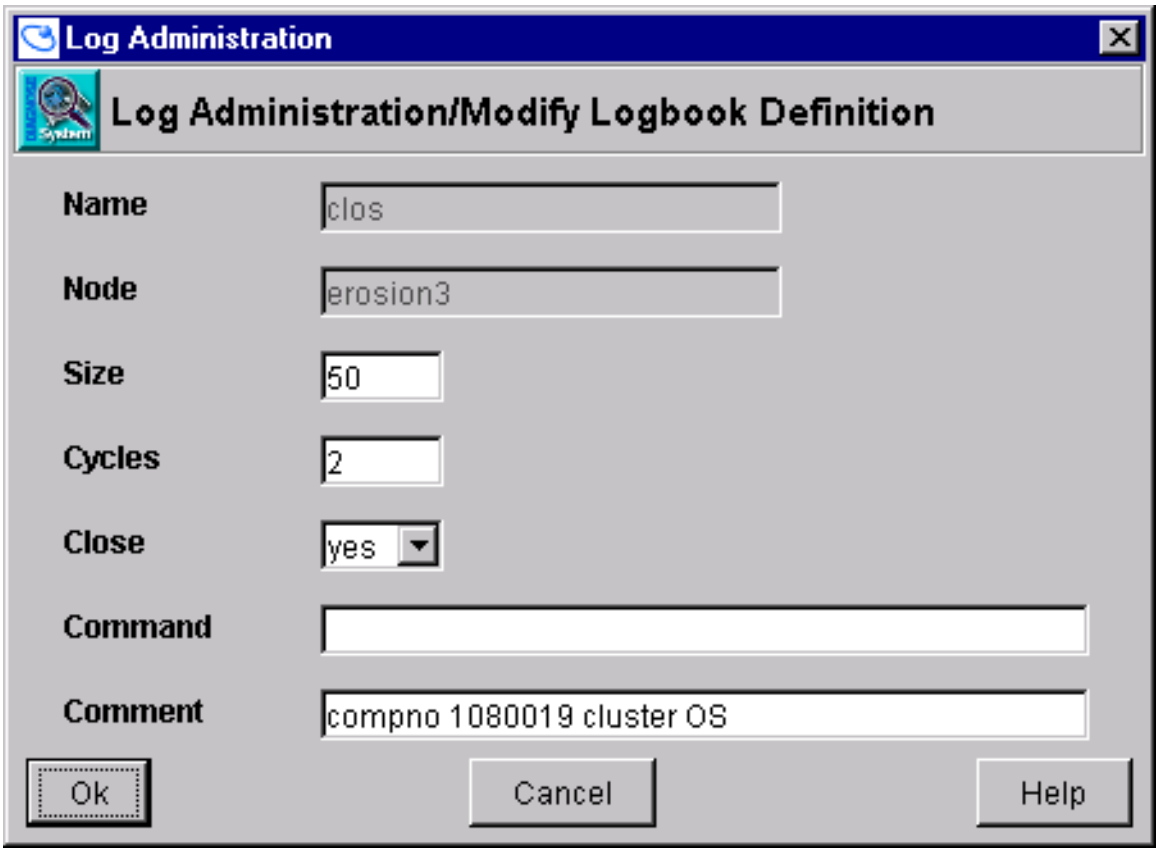


Figure 7: Dialog box for modifying a logbook definition

If you want to delete a logbook definition, you must first delete all component assignments for this definition.

Existing logbooks are not deleted if you delete a logbook definition. They can still be evaluated. If you no longer need these logbooks, you have to delete the relevant files in the base directory *$LOGPATH*.

Log entries which were due to be written to a logbook whose definition has been deleted are not lost. These entries are written to the *default* logbook.

4.1.5 Log entries

4.1.5.1 Evaluating log entries

You evaluate log entries from the *Log Evaluation* tab. They are evaluated automatically after the start of System Diagnostics.

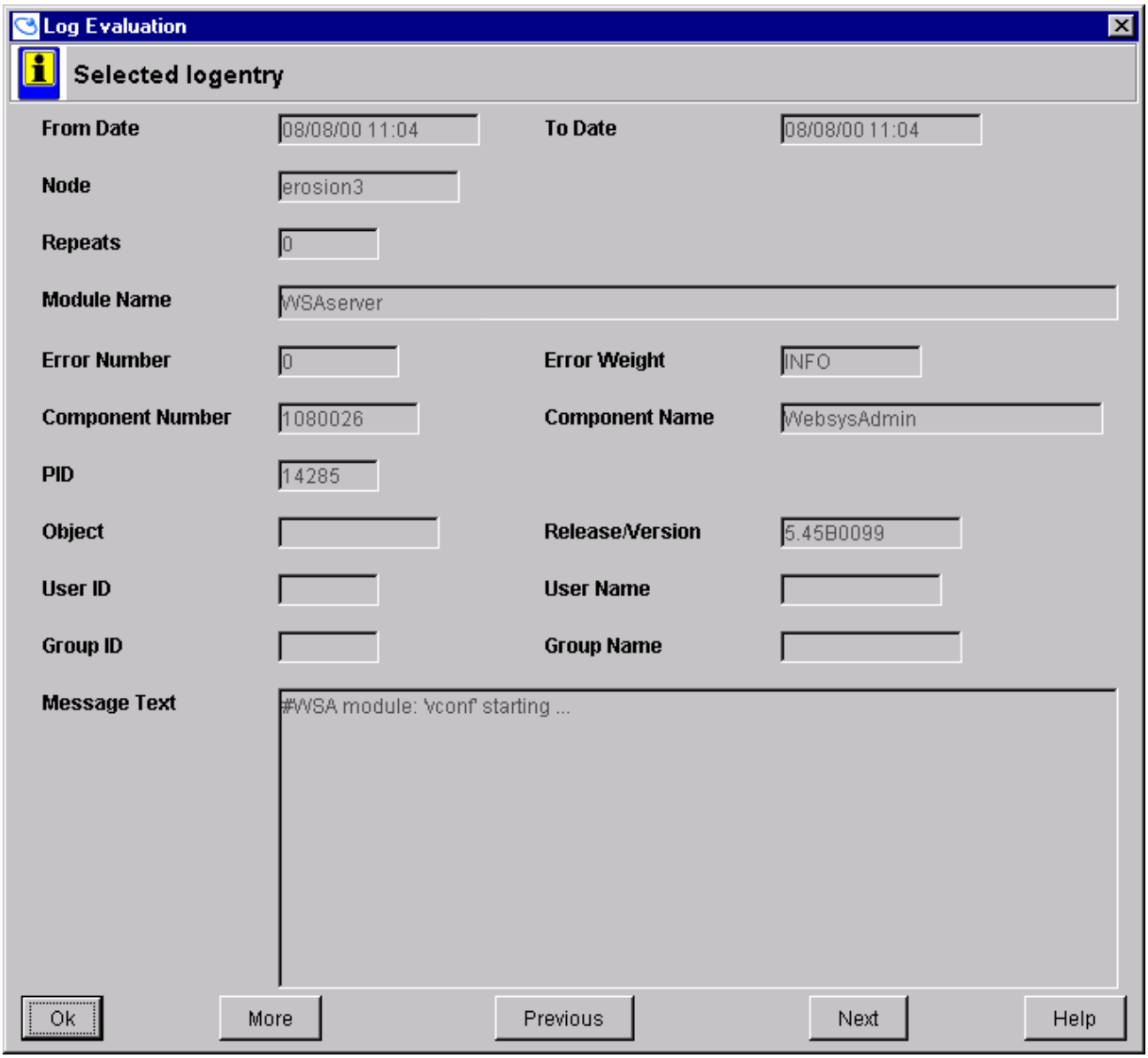


Figure 8: Dialog box for evaluating individual log entries

4.1.5.2 Selecting log entries for evaluation

You can narrow down the evaluation of log entries by means of the *Filter* action, or by selecting objects from the tool bar or by selecting nodes in the hierarchy browser.

Selecting log entries via a filter

From the *Log Evaluation* tab you can choose the *Filter* action to define criteria for the evaluation of log entries. This is the most comprehensive way of selecting log entries. The criteria that you define here apply for the duration of the current session at most.

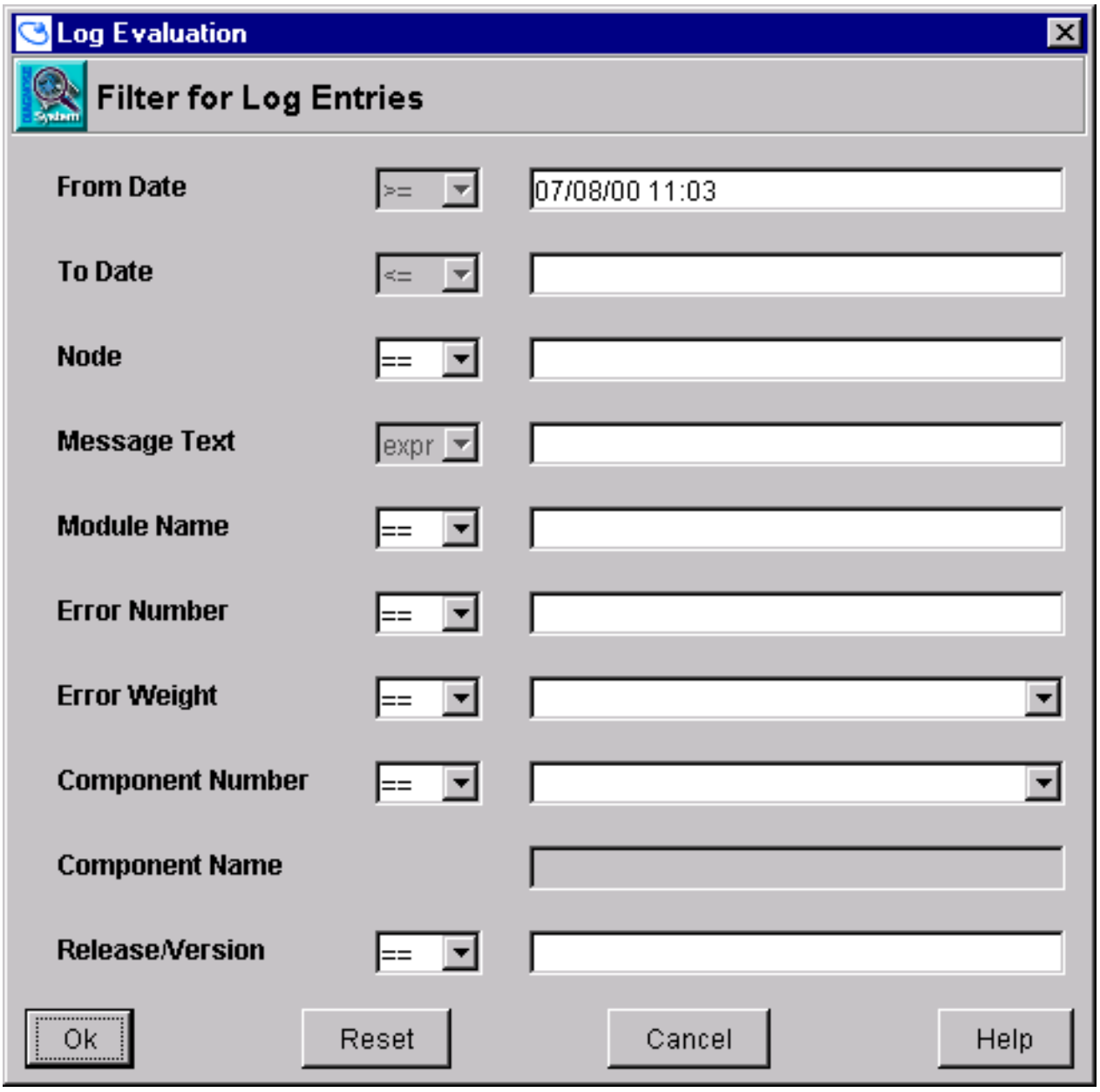


Figure 9: Dialog box for selecting log entries

Selecting objects from the tool bar

You can narrow down the evaluation of log entries by selecting objects from the tool bar. Like the *Filter* action this option offers you a choice of criteria. A dropdown menu opens from which you select an object to act as a

filter.

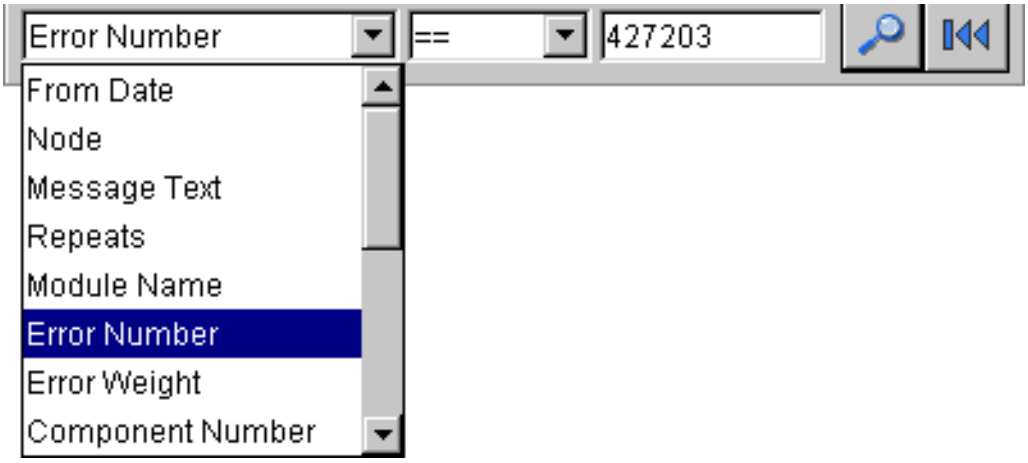


Figure 10: Selecting objects from the tool bar

Selecting nodes in the hierarchy browser

If you are in an administration domain you can narrow down the evaluation of log entries to a certain node by selecting the required node in the hierarchy browser. If you select the entire domain (cluster) then the log entries for all nodes on which Logging has been installed are taken into account.

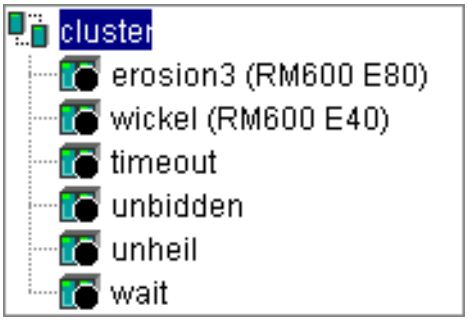


Figure 11: Selecting nodes in the hierarchy browser

4.1.5.3 Passing log entries to Event Management

Two conditions must be satisfied before log entries can be passed to Event Management:

- Event Management must have been started.

  Event Management is generally started automatically when the computer is booted and the logging server is started. If this is not the case you can start Event Management by setting the start option and restarting the logging server.
- In the course of component assignment you must have arranged for log entries relating to this component to be passed on to Event Management (...).

You can see whether the start option has been set and whether Event Management is running in the *Event Management* field on the *Status* subtab of the *Log Administration* tab. The value *yes* or *no* is output for both items of information in this field. The two items are separated by a colon, e.g.:

yes:yes Event Management is supposed to run and is running.

yes:no Event Management is supposed to run but is not running.

4.1.5.4 Storing log entries in an ASCII log file

All log entries for which you have defined that they are due to be passed on to an administration tool such as Unicenter TNG are not only written to logbooks but are also stored in a procol file. There is such a protocol file for every node in an administration domain.

Passing on is controlled

- either via a reaction for individual log entries in Event Management
- or via component assignment for all log entries (...).

In the latter case the logging server must also be started with the *Entries to ASCII protocol* start option. You can see whether the logging server was started with this option and whether entries are being stored in a protocol file in the *ASCII Protocol* field on the *Status* subtab of the *Log Administration* tab. The value *yes* or *no* is output for both items of information in this field. The two items are separated by a colon, e.g.:

yes:yes Log entries are supposed to be stored in a protocol file and are being written there.

yes:no Log entries are supposed to be stored in a protocol file but are not being written there.

You administer the protocol file by opening the *Log Administration* tab followed by the *Protocol File* subtab.

4.1.6 Event Management

The Event Management facility enables you to define events on the basis of errors and their frequency, and to trigger reactions to them.

An event occurs when an error occurs so frequently in a certain period that the threshold that has been specified in the event definition is reached.

Events and reactions are described in event and reaction definitions. You can create your own event and reaction definitions and edit them. You cannot edit event and reaction definitions that Fujitsu Siemens has supplied with the system as they are protected to make sure they are not modified or deleted by mistake. Their attributes are based on experience and should not be modified as a rule. If it turns out to be essential to modify protected definitions please get in touch with the Operating System Development department at Fujitsu Siemens (stating *Diagnostics* or *Logging* as the subject).

4.1.6.1 Administering event definitions

You administer event definitions by opening the *Event Management* tab followed by the *Events* subtab.

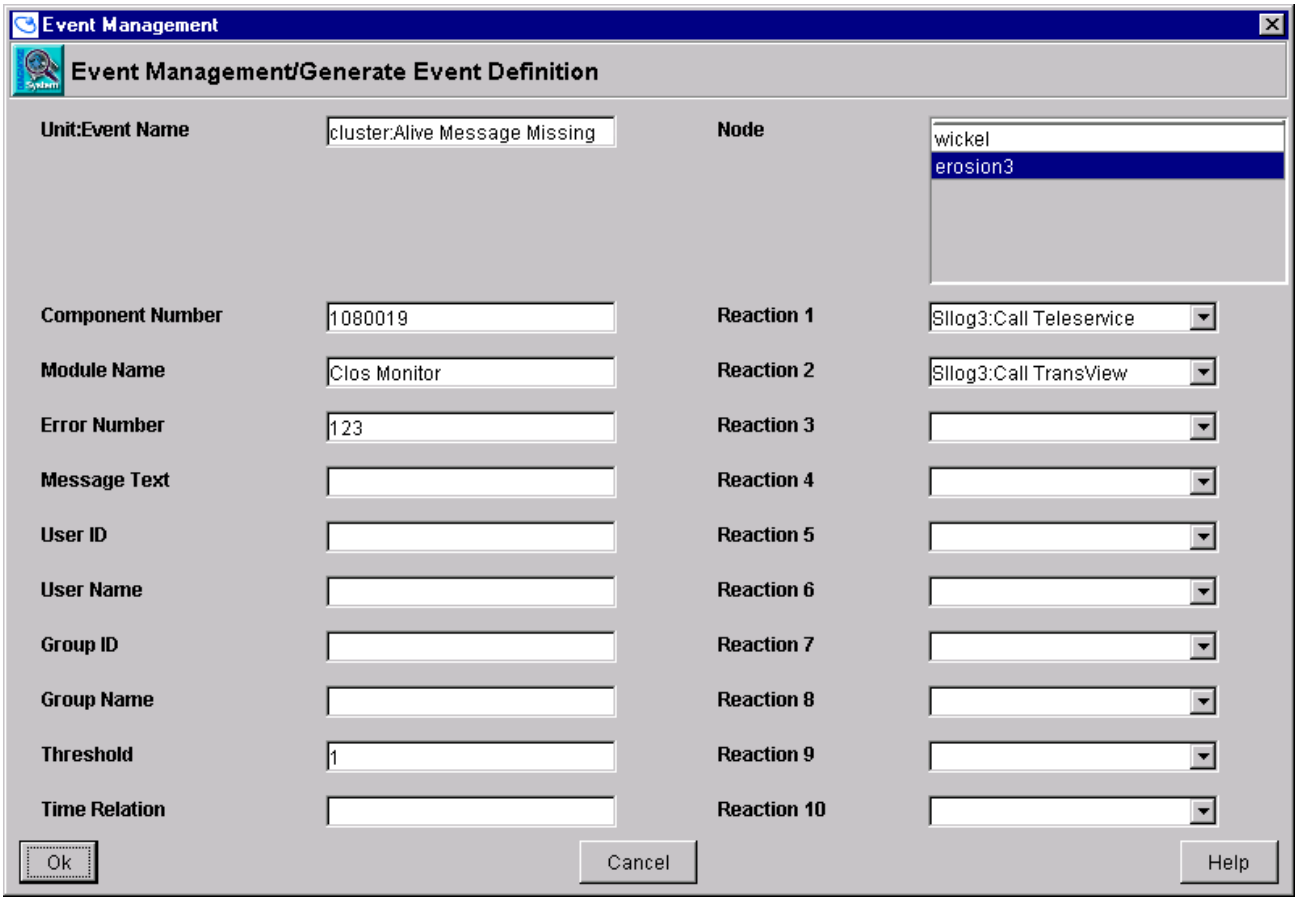


Figure 12: Dialog box for creating an event definition

If you delete an event definition this event will no longer occur, and consequently any reaction to it that you have defined will not take place either.

4.1.6.2 Administering reaction definitions

You administer reaction definitions by opening the *Event Management* tab followed by the *Reactions* subtab.

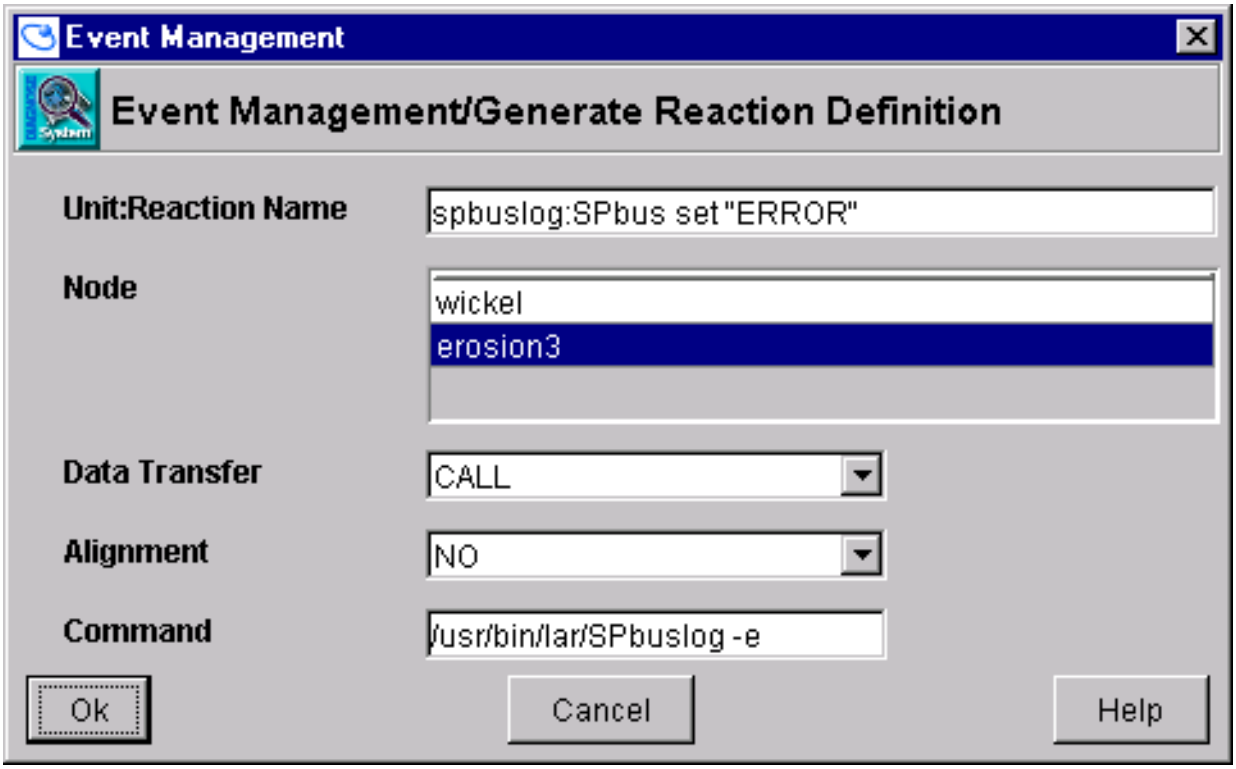


Figure 13: Dialog box for creating a reaction definition

If you delete a reaction definition which is referenced in an event definition this reaction can no longer be triggered.

4.1.6.3 Administering counters

You administer counters by opening the *Event Management* tab followed by the *Counters* subtab.

Counters provide information on the following:

- How often a log message has occurred in a certain period (counter) and how close this has taken you to a defined threshold. The period and threshold are specified in the relevant event definition.
- How often a threshold has been reached (transgression), thus giving rise to an event.

You can initialize the *Counter* and *Transgression* fields via the *OK* button.

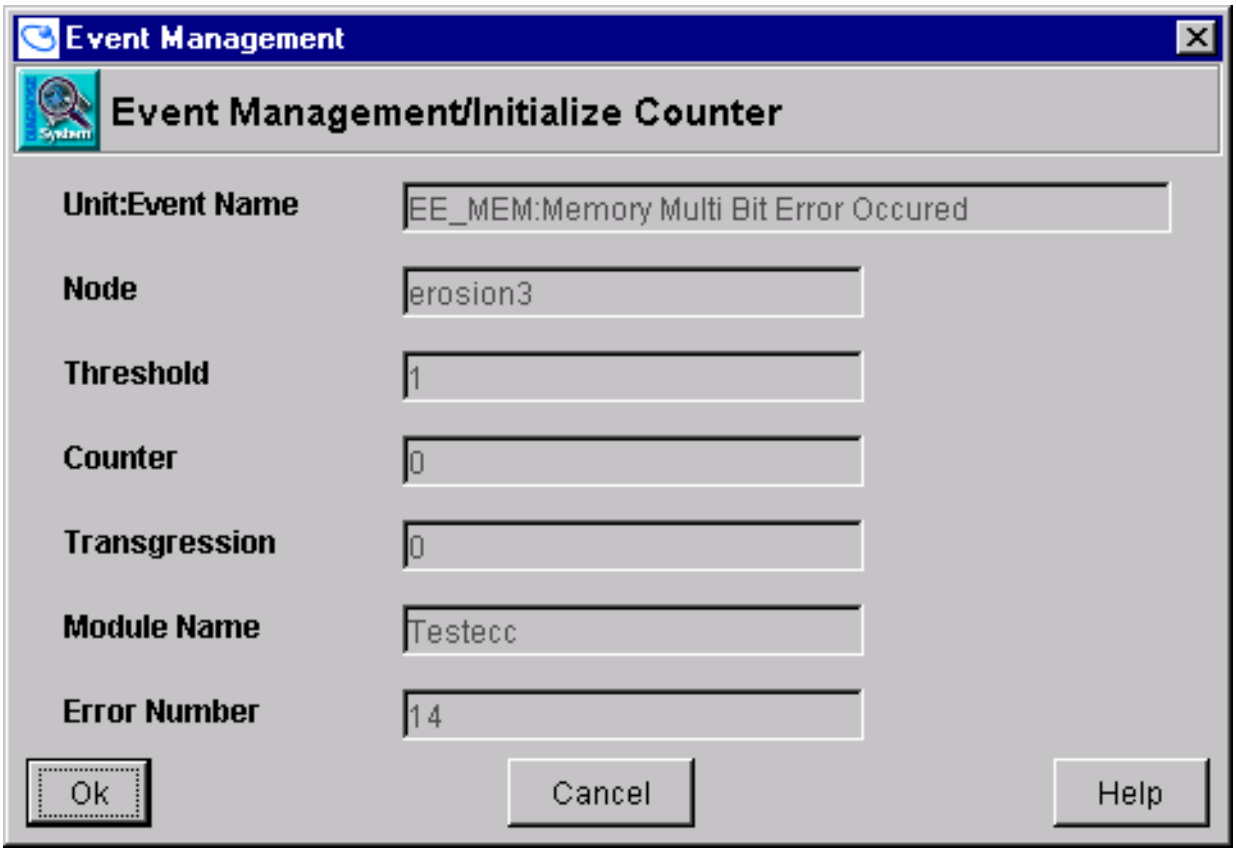


Figure 14: Dialog box for initializing counters

4.1.7 Threshold Monitoring

You can edit threshold tables for SCSI devices, memory boards and caches on CPU boards on the *Monitoring* tab.

The aim of monitoring is to detect errors as early as possible, in other words when the hardware is still functioning overall but a critical error limit has been reached which is no longer acceptable from the point of view of data security.

Threshold tables contain the parameters required for monitoring the devices or boards. You will find the actual values for these parameters in the technical specifications supplied by the respective hardware vendors.

4.1.7.1 Checking thresholds

The error log daemon makes a distinction between *soft limits* and *hard limits* in the threshold values. When one of these threshold values is reached, an error message is sent to Logging, which records it and sends an event report to Event Management. If necessary, Event Management will initiate a reaction (see also Section "Event Management"). In addition to this, the device or board will be recorded in the HW_CONFIG database and flagged with status *error* (for the soft limit) or *defect* (for the hard limit) (... (RM600)(RM600) or (RM200/300/400) ... (RM200/300/400)).

4.1.7.2 Administering threshold tables

You administer threshold tables by opening the *Monitoring* tab and the corresponding subtab.

You can only edit threshold tables that you have created yourself. Threshold tables that have been supplied by Fujitsu Siemens are protected and cannot be modified.

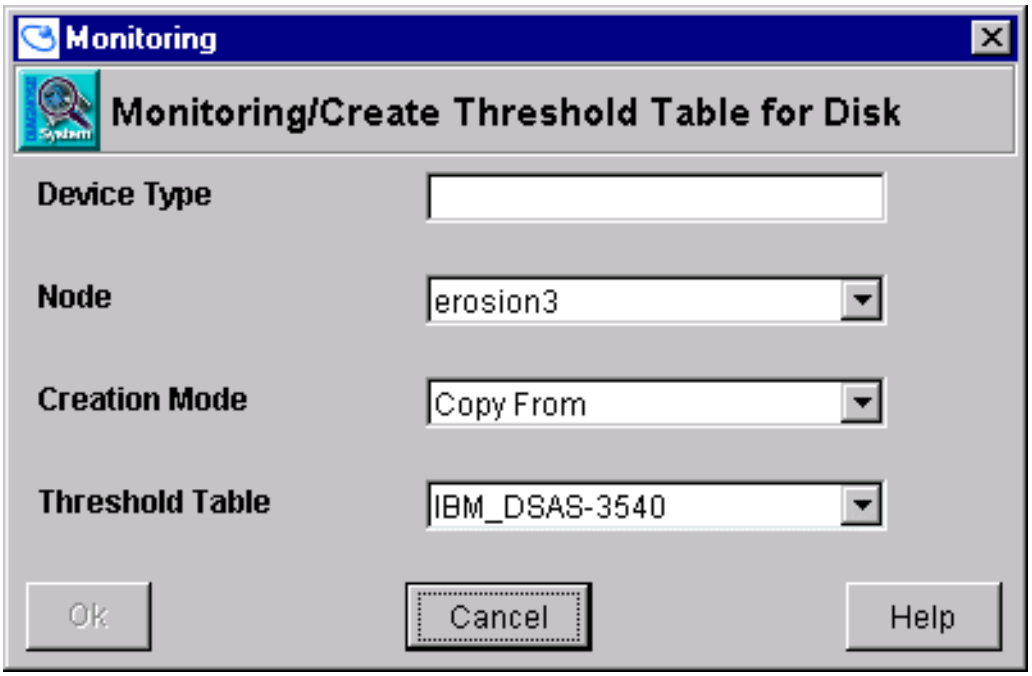


Figure 15: Dialog box for creating a threshold table

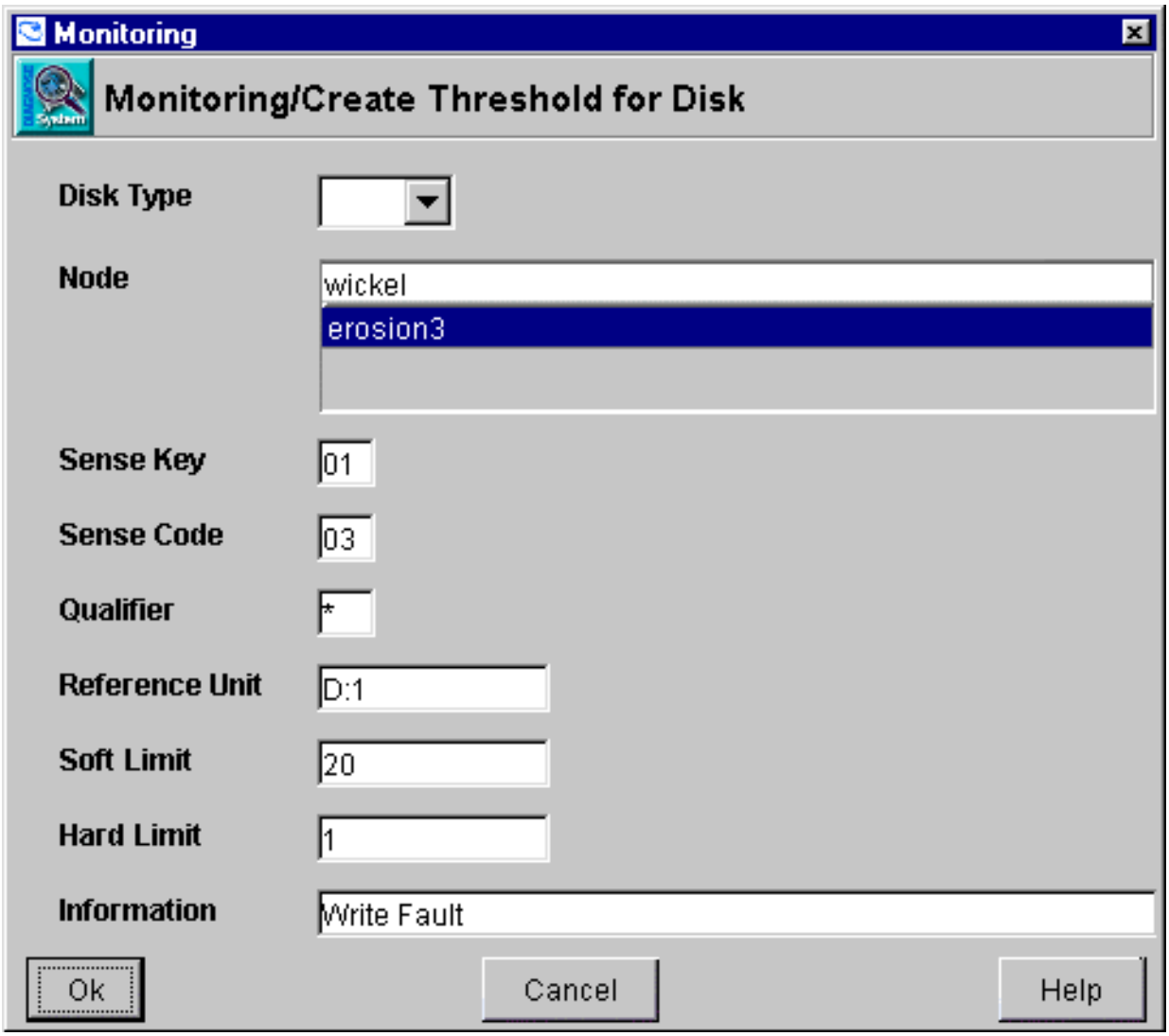


Figure 16: Dialog box for creating an entry in a threshold table

4.2 System level

This chapter is aimed at programmers who want to use Logging from their own applications or want to extend the properties of Logging. It gives information on:

- Selection of the interface to be used for sending messages to Logging
- Structure of log entries
- Structure of logbooks and log files
- Logging administration files
- Displaying additional information when evaluating log entries
- Extending Event Management

4.2.1 Interfaces

Messages can be passed to Logging in different ways. A C language application programming interface (API) named Logging V3.0 Library and the command *log3logger*(1M) are available as direct interfaces, and the C routine *syslog*(3C) and the command *logger*(1-ucb) as indirect interfaces.

Normally the direct way of passing messages to Logging should be used. Under certain conditions, though, it may be necessary to pass messages to Logging indirectly, e.g. for applications that are closely related to the operating system.

4.2.1.1 Passing data directly to Logging

This section describes the C language API with which communication between applications and Logging can be implemented directly. You can find more information on the *log3logger* command, which also supports direct communication, on the manual page of the same name.

The API has been implemented in the libraries *$LOGPATH/lib/liblog3.a* and *$LOGPATH/lib/liblog3.64s.a* (64-bit version). The associated file named *logging.h* is stored in the directory *$LOGPATH/include*.

The following functions are available for communication between an application and Logging:

- First the pipe to Logging is opened with the function *log3open*(5). It is not necessary to open the pipe for every log entry and then close it again. The pipe should remain open until the last log entry has been sent.
- Then log data can be written to a buffer with the functions *log3printf*(5) and *log3memcpy*(5). A log record is placed in the buffer whenever one of these functions is called. However it is only necessary to split up log data into individual log records when you want to structure the data and add guidance texts to them.

  The buffer size is confined to 2 KB.
- When all log records for a log entry have been placed in the buffer they are passed to Logging via the function *log3write*(5). Whenever this function is called a log entry is created by adding a log data header to the log records. If you want to structure the log entry you can pass a structure number as a parameter with the data (see Section "The log3struct file").
- The pipe to Logging is closed with the function *log3close*(5).

  The pipe is closed automatically if an error occurs in the *log3write* function.

4.2.1.2 Passing data indirectly to Logging

In addition to passing log data to Logging directly it is possible to pass it indirectly with the aid of the *syslog* routine or the *logger* command via the Syslog daemon (see *syslogd*(1M)).

This way can be used by the operating system or applications closely related to it in order to document any problems which occur, for instance, while the system is booting before Logging is started, or which occur when Logging is not started when booting in single-user mode. Generally, messages which occur between booting and the start of the Syslog daemon are not lost because the Syslog daemon first reads the kernel message buffer when it starts.

Messages which are passed to Logging via the system logging facility do not normally contain a log data header that can be evaluated by Logging. The message can be given an evaluable header by placing the header data in front of the message text before the message is passed to the *syslog* routine (see *log3header*(5)).

The following example shows an application in which header data is placed in front of the message text before the message is passed to the *syslog* routine.

```
#include <syslog.h>
main()
{
□□□char□□□□□□□□□□□□*log3header();
□□□unsigned□long□□compno;
□□□unsigned□short□errno,□level,□structid;
□□□char□□□□□□□□□□□□*rel_ver,□*ident,□*head;
□□□compno□□□=□123456789L;□□□□/*□Component□number□□□□□□□□□□□□□□□□*/
□□□errno□□□□=□100;□□□□□□□□□□□□/*□Error□number□□□□□□□□□□□□□□□□□□□□*/
□□□level□□□□=□LOG_WARNING;□□□/*□Error□weight□□□□□□□□□□□□□□□□□□□□*/
□□□structid□=□123;□□□□□□□□□□□□/*□Structure□number□□□□□□□□□□□□□□□□*/
□□□rel_ver□□=□"5.45B00";□□□□□/*□Release/version□of□component□□□*/
□□□ident□□□□=□"xyz";□□□□□□□□□/*□Component□name□□□□□□□□□□□□□□□□*/
□□□head□=□log3header(compno,□errno,□level,□structid,□rel_ver,□ident);
□□□...
□□□openlog("",□0,□LOG_USER);
□□□syslog(LOG_WARNING,□"%sfile system□%s□is□full",□head,□"/usr");
□□□...
}
/*□Subroutine□to□create□the□header□*/
char□*log3header(compno,□errno,□level,□structid,□rel_ver,□ident)
unsigned□long□□compno;
unsigned□short□errno,□level,□structid;
char□□□□□□□□□□□□*rel_ver,□*ident;
{
□□□static□char□buffer[80];
□□□long□□□□□□□□now
□□□(void)□time(&now);
□□□(void)□sprintf(buffer,□"LOG3.0%-10lu%-10lu%-5u%-5u%-5u%-12.12s%-16.16s",
```

```
□□□□□□□□□□□□□□□□□□□□now,□compno,□errno,□level,□structid,□rel_ver,□ident);
□□□return(buffer);
}
```

4.2.2 Structure of a log entry

A log entry consists of a log data header, one or more log records containing the message text, and two bytes each at the beginning and end containing the total length of the log entry. The bytes marked (*) belong to the log data header.

The log data header contains information about a log entry. It is used to identify the log entry during evaluation. All header data, apart from the first two bytes which are not evaluated, can be used for selection purposes. The most important information in the log data header is output as well when log entries are evaluated.

Byte	Meaning/explanation
0-1	Total length of the log entry in bytes The following rules apply to the total length: – The value of the *LOGSIZE* constant which is defined in the *logging.h* file must not be exceeded (at present 2 KB). – The total length is increased to a fourfold of four bytes.
2*	System platform identifier (*0x38* for RM computers)
3*	Operating system identifier (*L* for Reliant UNIX)
4-7*	Component number Fujitsu Siemens gives every component an internationally unique number ranging from *1* to *4294967295*. This number is used to assign components to logbooks. The component number is passed to Logging as a parameter of the *log3open* function and can be used as a selection criterion for evaluation. If you want to assign numbers to your own components, please get in touch with the Operating System Development department at Fujitsu Siemens (stating Logging as the subject) so that you can get numbers that have not yet been assigned.
8-9*	Error number of the module Error numbers ranging from *0* to *65535* can be defined for certain log data in every module of a component. The error number can be used as a selection criterion during evaluation. It is passed to Logging as a parameter of the *log3write* function.
10-11*	Repeat counter If repeat detection has been enabled these bytes contain the number of times the log entry as been written in direct succession to the same logbook. The counter is stopped at *65535* in the case of an overflow.
12-15*	Time stamp of the first log operation The time the log entry was first written is entered here.
16-19*	Time stamp of the last repeat The time the log entry was last written is entered here.
20-21*	Error weight The error weight indicates the severity of the error. Eight levels (*0* to *7*) are predefined in the *logging.h* file. Other levels from *8* to *65535* can be freely defined. The error

	weight is passed to Logging as a parameter of the *log3write* function. List of predefined error weights: 0 = EMERG — System halts. Generally an error with this weight can no longer be logged. 1 = ALERT — Errors indicating considerable loss of data and requiring immediate action. 2 = CRIT — Errors indicating loss of data or malfunctions requiring action soon. 3 = ERR — Errors indicating malfunctions but not requiring immediate action. 4 = WARNING — Warnings that normal functioning of the system could be impaired unless appropriate action is taken. 5 = NOTICE — Notices of potential problems. 6 = INFO — Status messages. 7 = DEBUG — Messages only used in program debugging.
22-23*	Structure number Structure numbers describe the structure definitions in the *log3struct* file and are used for presentation purposes when log data is evaluated. The structure number is passed to Logging as a parameter of the *log3write* function. It can range from *1* to *65535*.
24-25*	ID of the process that generated the log entry
26-27*	Length of the module name including delimiter (\0)
28-39*	Release/version of the component The release and version of the component is passed to Logging as a parameter of the *log3open* function.
40-55*	System ID This parameter contains the system name.
56-<*n*>*	Module name including delimiter (\0) The module name is passed to Logging as a parameter of the *log3open* function.
<*n+1*>-<*m*>	Log records
<*m+1*>-<*m+2*>	Total length of log entry in bytes

4.2.3 Logbooks and log files

Log entries are stored in logbooks which are normally located in the directory defined by *$LOGPATH* (*/opt/log3* by default). Administration of logbooks is controlled via the file *log3admin*(4).

Every logbook consists of a cycle of two to nine separate log files which have the suffix *[0-8]log*. The log files are created one after another when they are needed to store data. When a log file is full the next one is created. When the cycle has been completed, the oldest log file is overwritten.

Whenever a log file is created it is given a log file header which is updated with every log operation. The log file header contains information about the log file that is needed during evaluation.

Structure of the log file header

Byte	Meaning/explanation
0-3	Logging password These bytes contain the constant *LOGG*.
4-5	Logging release The release number has been defined as *30* at present.
6-7	Length of the log file header in bytes This value defines the position of the first log entry in the file (byte 276 in the log file).
8-23	System ID As long as there is no internationally valid system ID this parameter will contain the computer's node name.
24-55	Operating system name This parameter is ascertained with the *uname* command (without any option).
56-87	Computer's node name This parameter is ascertained with the *uname -n* command.
88-119	Operating system release This parameter is ascertained with the *uname -r* command.
120-151	Operating system version This parameter is ascertained with the *uname -v* command.
152-183	Computer type This parameter is ascertained with the *uname -m* command.
184-215	Log file name
216-247	Command If a command is specified in the *log3admin* file which has to be executed when the log file has reached its maximum size, then it is taken from *log3admin*.
248-251	Maximum file size This parameter is taken from the *log3admin* file.
252-255	Current file size This parameter is updated after every log operation.
256-259	Time stamp of the first log entry in the log file
260-263	Time stamp of the last log entry in the log file
264-267	Number of entries in the log file
268	File full

	After every log operation *n* (no) is entered here unless the log file has reached its maximum size. In that case *f* (full) is entered here.
269	File being processed □ If the file is open in write mode this byte contains *y* (yes). Otherwise it contains *n* (no).
270	Number of files (cycles) in a logbook □ This parameter is taken from the *log3admin* file.
271	Leave file open after log operation or close it □ This parameter is taken from the *log3admin* file.
272-275	File descriptor□ The file descriptor is stored here if the log file is open, otherwise the value *-1*.

4.2.4 Administration files

4.2.4.1 The log3admin file

Logging is controlled with the *$LOGPATH/log3admin* file. This file contains four blocks:

- The properties of the logbooks are described row by row in the *LOGFILE* block. The columns have the following meaning:

  FILENAME
  : Logbook name

  SIZE
  : Size of the log files in a logbook in KB

  CYCLE
  : Number of log files in a logbook (2-9)

  CLOSE
  : Flag indicating whether the current log file of a logbook is to remain open (0) or be closed (1) after a log operation

  COMMAND
  : Command to be executed when a log file has reached its maximum size (optional)

  COMMENT
  : Optional comment

```
#LOGFILE□
;FILENAME□□□□□□□SIZE□□□□□□CYCLE□CLOSE□□□□□□□□□COMMAND□□□□□□COMMENT□
;-------------□□---------□-----□-------------□----------□□□----------------□
□
datasyslog□□□□□□1000□□□□□□4□□□□□0□□□□□□□□□□□□□□□□□□□□□□□□□□□□;□system□logbook□
...□
```

- The *LOGTYPE* block defines, row by row, the logbooks which the log entries for the respective component number are to be written to. It also defines whether log entries are to be passed on and, if so, where to. The columns have the following meaning:

  COMP NO
  : Number of the reporting component (an asterisk (*) at this point means "all components")

  LOGFILE
  : Logbook to which the component writes (an asterisk (*) at this point means "all logbooks")

  REPEATS
  : Repeat counter enabled (1) or not (0)

  DIRECTION
  : Flag indicating whether the log entry is to be passed on and, if so, where to

  COMP NAME
  : Name of the reporting component as an optional comment

```
#LOGTYPE□
;COMP□NO□□□□LOGFILE□□□□REPEATS□□DIRECTION□□□□□□□□□□□□□□□□□□COMP□NAME□
;----------□----------□--------□-------------------------□-----------------□
□
1□□□□□□□□□□□□datasyslog□1□□□□□□□□□6□□□□□□□□□□□□□□□□□□□□□□□□□□□□;□Logging□V3.0□
...□
```

  Every component that is entered writes to the assigned logbooks. If a component has not been assigned to any logbook, or if the specified logbook does not exist, the log entries are stored in the default logbook.

- The properties of the default logbook are defined in the *DEFAULT* block. The syntax is identical to that of the *LOGFILE* block, but this block only contains one entry and that must be present. All parameters in this entry, including the file name, can be modified.

```
#DEFAULT□
```

```
;FILENAME□□□□□□□□SIZE□□□□□□□CYCLE□CLOSE□□□□□□□□□□COMMAND□□□□□□□COMMENT□
;-------------□□---------□-----□-------------□----------□□□----------------□
□
default□□□□□□□□□□1000□□□□□□□9□□□□□□0□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□;□default□logbook□
...□
```

- The properties of the ASCII log file are defined in the *PROTOCOL* block. The syntax is identical to that of the *LOGFILE* block, but this block can only contain one entry, and only the *SIZE* parameter can be modified.

```
#PROTOCOL□
;FILENAME□□□□□□□□SIZE□□□□□□□CYCLE□CLOSE□□□□□□□□□□COMMAND□□□□□□□COMMENT□
;-------------□□---------□-----□-------------□----------□□□----------------□
□
protocol□□□□□□□□□1000□□□□□□□2□□□□□□1□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□;□ASCII□...□
...□
```

4.2.4.2 The log3struct file

The *$LOGPATH/log3struct* file contains structure definitions (layout descriptions) for log records. These structure definitions can be used for presentation purposes when the log entries are evaluated.

A unique structure number ranging from *1* to *65535* is assigned to every structure definition. When a log record is written, the structure number is passed as a parameter of the *log3write* function or of the *log3logger* command.

If no structure number or a non-existent one is passed when log records are written - or if *0* is passed as the structure number - then every log record is output in a new row without a guidance text.

Structure definitions consist of guidance texts for the individual records of a log entry. During evaluation these guidance texts are placed in front of the log records. The guidance texts are specified as strings and may contain formatting characters such as *\n*, *\r*, *\t*, *\b*, *\000* to *\377*.

If a log entry consists of more records than guidance texts in the structure definition, then the remaining log records are output without any guidance texts. If a structure definition contains more guidance texts than records in the log entry, then the excess guidance texts are appended to the end of the output.

4.2.5 Displaying additional information

Additional information can be output when log entries are evaluated, for example to explain the problem in more detail and show potential solutions. Shell scripts or C programs are called from defined directories to do this. The output of the shell scripts and C programs that you create in these directories must be sent to the standard output (stdout).

The following call parameters are available for shell scripts and C programs. These parameters can be evaluated, for example to react differently to error numbers. The parameters are always passed in the following order:

- Repeat counter (from log entry, bytes 10-11)
- Date of first log entry (from log entry, bytes 12-15)
- Time of first log entry (from log entry, bytes 12-15)
- Date of last log entry (from log entry, bytes 16-19)
- Time of last log entry (from log entry, bytes 16-19)
- Component number (from log entry, 4-7)
- Component name (from the *log3admin* file)
- Error number of module (from log entry, bytes 8-9)
- Error weight (from log entry, bytes 20-21)
- Process ID (from log entry, bytes 24-25)
- Release/version of component (from log entry, bytes 28-39)
- System ID (from log entry, bytes 40-55)
- Module name (from log entry, bytes 56-<*n*>)
- Log records (from log entry, bytes <*n+1*>-<*m*>)

There are two ways of displaying additional information:

- Automatic display

  With automatic display, the additional information is output automatically when a single log entry has been selected from the chronological list of entries and has been displayed.

- Display on demand

  With display on demand, the additional information is shown in a separate text window when the More button has been pressed.

There are three kinds of additional information. The shell scripts or C programs containing or generating the additional information must be located in the following directories which depend on the type of display:

Additional information of a general nature
: Automatic display: *$LOGPATH/evaluations/general*
  Display on demand: *$LOGPATH/evaluations/button/general*

Additional information depending on the component number
: Automatic display: *$LOGPATH/evaluations/<component_number>*
  Display on demand: *$LOGPATH/evaluations/button/<component_number>*

Additional information depending on the module name
: Automatic display: *$LOGPATH/evaluations/button/<module_name>*
  Display on demand: *$LOGPATH/evaluations/button/<module_name>*

4.2.6 Extending Event Management

When you have implemented a log interface in one of your applications you can associate reactions with the occurrence of log messages. You do this by first defining events in event description files. An event occurs when a message is generated n times within a specific period and certain selection criteria are met.

You can react to every event that has been defined in this way by calling up to ten commands. These reactions are defined in reaction description files.

Event and reaction definitions can be edited directly in the description files both via the *WebSysAdmin* user interface and on the system level.

You can modify the event and reaction description files that are supplied by Fujitsu Siemens directly on the system level, or you can create new files, or assign files from other packages in the installation process. Note, however, that the files supplied by Fujitsu Siemens are overwritten in the course of an update installation and consequently any modifications would be lost. It is therefore advisable to store your own events and reactions in your own files.

Reaction programs are launched with the aid of the defined reactions. These reaction programs must comply with certain rules concerning call syntax, passing of log information and general behavior.

4.2.6.1 Editing event description files

Event description files are located in the *$LOGPATH/filters* directory and are named *events.<suffix>*. When you create a file of your own you should choose a suffix that ensures it will not be used elsewhere. The provision of an event description file with the same suffix from somewhere else will cause the existing file to be overwritten. An event description file can contain multiple event descriptions. Each new event description begins with the keyword *EVENT:*.

An event description requires an event name, a filter criterion, a limit (greater than zero) and a reaction call at least. Incomplete event descriptions are accepted but the event may never occur in certain circumstances with the result that no reaction would ever by triggered.

Structure of an event description (example with explanations):

Keyword	Parameter
EVENT:	NAME=Fan Failure Name of the event description. The name may contain space characters but no colon (:).
FILTER:	COMPONENT=4294967292 Number of the reporting component ranging from 1 to 4294967295. MODUL=Logging Server Name of the reporting module. The name may contain space characters but no colon (:). ERROR=2,5-9,23 List of the error numbers of the module ranging from 0 to 999999. Individual numbers, numbers separated by commas and ranges of numbers can be specified. TEXT=^Printer lsr3 is out of paper!$ Regular expression containing the message text.
THRESHOLD:	LIMIT=1 Number of messages ranging from 0 to 65535. When this limit is reached, an event has occurred in connection with the following parameter. TIMERELATION=D:1 Period in which the occurrence of a message is counted. This parameter has the following format: [D|H|M|S]:<*positive number*>. The letters *D*, *H*, *M* and *S* stand for *day*, *hour*, *minute* and *second*.
REACTION:	ACTION=SIlog3:Console ACTION=SIlog3:Call Config ACTION=SIlog3:CALL Teleservice Reactions (max. 10) that are called when an event occurs. This parameter has the following format <*suffix of reaction file*>:<*reaction name*>. The name may contain space characters but no colon (:).

Table 1:

4.2.6.2 Editing reaction description files

Reaction description files are located in the *$LOGPATH/filters* directory and are named *reactions.*<*suffix*>. When you create a file of your own you should choose a suffix that ensures it will not be used elsewhere. Provision of a reaction description file with the same suffix from somewhere else will cause the existing file to be overwritten. A reaction description file can contain multiple reaction descriptions. Each new reaction description begins with the keyword *ACTION:*.

Reaction programs are launched with the aid of reactions that have been defined in this way. These reaction

programs must comply with certain rules concerning call syntax, passing of log information and general behavior.

Structure of a reaction description (example with explanations):

Keyword	Parameter
ACTION:	NAME=SIlog3:Console The reaction is referenced under this name in event descriptions. This parameter has the following format: *<suffix of reaction file>:<reaction name>*. The name may contain space characters but no colon (:).
	ALIGNMENT=YES Starting time of the next reaction (*YES* or *NO*). *YES* means that the next reaction will not start until this one has finished or not until after 90 seconds.
	DATATRANSFER=CALL Mode of passing log information to the reaction program. *CALL* means that the information is passed as a call. *PIPE* means the information is passed via the standard input.
	COMMAND=console -c Call of the reaction program. Wildcards and I/O redirection are not allowed, but space characters and environment variables are.

Table 2:

4.2.6.3 Activating event descriptions

New or modified event descriptions have to be activated with the command *log3do -c logeventsact* before they take effect.

Restarting Logging and rebooting the system do not activate new or modified event descriptions!

Reaction descriptions do not need to be activated. Only when an event has occurred are the reaction descriptions searched for and the found reactions executed. All changes to reaction files are thus taken into consideration when the next event occurs.

4.2.6.4 Creating reaction programs

When an event occurs the defined reactions are searched for in the reaction description files and the associated reaction programs are executed. If no complete path names have been specified for the reaction programs, this means that the programs are located in the *$LOGPATH/reactions* directory.

In addition to the parameters that were passed in the reaction program call, the reaction program can also access the complete log information in the log entry that triggered the event. The *LIMIT* and *TIMERELATION* parameters from the event description and an object identifier are also passed. The reaction program can use this information to control the necessary reaction.

Passing log information to reaction programs

Log information (not log entries as such) can be passed to a reaction program via the standard input (*DATATRANSFER=PIPE*) or as call parameters (*DATATRANSFER=CALL*). The *DATATRANSFER* keyword defines the transfer mode in the reaction description. The particular reaction program must be adapted to the transfer mode that has been defined. The standard input has to be used when the log information contains binary values.

When log information is passed via the standard input, the reaction program receives it in raw format. The standard input can only be evaluated by a C program that is aware of the exact structure of log entries. The log information has to be converted into ASCII for evaluation purposes. This is done by calling the *log3trans* routine.

When the log information is passed as call options the reaction program receives it in the shape of a parameter list which is always appended to the call for the reaction program. The parameter list is always passed completely as

-X <ParameterList> -X

and in the same order. All parameters in this list are passed as ASCII strings. Null strings are passed as placeholders for empty parameters.

Sample reaction program

The following example shows a shortened version of the sample reaction program named *example* which you can find in the *$LOGPATH/reactions* directory. This reaction program stores the passed parameters in the */tmp/reaction$$* file. Depending on the call, either the standard input or the parameter list of option *-X* is evaluated.

Calling the reaction program by Event Management:
example [<*parameters*>] -X <*parameter_list*> -X

parameters
: Parameters are any program-dependent options and call parameters with an option identifier.In the example, *PIPE* can be specified as a parameter if evaluation is intended to take place via the standard input.

-X parameter_list -X
: The Event Management parameter list is appended automatically when the call is issued.

Structure of the reaction program:

```
#□Evaluation□of□the□standard□input□by□the□C□program□log3trans,□if□PIPE□is
#□is□specified□as□the□first□parameter.□The□log3trans□program converts□the
#□contents□of□the□log□entry□to□ASCII.□This□program is□stored□in□the
#□/opt/log3/bin directory.

if□[□"$1"□-eq□"PIPE"□]
then
□□□□□log3trans□>>□/tmp/reaction$$
□□□□□exit□0
fi

#□Evaluation□of□the□-X□parameter□list□if□the□PIPE□parameter□is□not□specified.

FILE□=□/tmp/reaction$$□export□FILE
date□>>□$FILE
echo□>>□$FILE
echo□"$*"□>>□$FILE
while□[□"$1"□!=□"-X"□];□do
□□□□□shift
done

shift
[□$#□-lt□2□]□&&□echo□"No□options!!"□>>□$FILE□&&□exit□0
echo□"limit□□□□□□□□□□□□□=□$1"□□□□>>□$FILE
echo□"time□□□□□□□□□□□□□□=□$2"□□□□>>□$FILE
echo□"object□□□□□□□□□□□=□$3"□□□□>>□$FILE
echo□"component no.□□□=□$4"□□□□>>□$FILE
echo□"error no.□□□□□□□=□$5"□□□□>>□$FILE
echo□"level□□□□□□□□□□□□=□$6"□□□□>>□$FILE
echo□"PID□□□□□□□□□□□□□□=□$7"□□□□>>□$FILE
echo□"release/version□=□$8"□□□□>>□$FILE
echo□"system name□□□□□=□$9"□□□□>>□$FILE
echo□"modul name□□□□□□=□{$10}"□>>□$FILE
echo□"struct no.□□□□□□=□{$11}"□>>□$FILE
```

```
shift□10
i□=□0
while□[□$#□-gt□1□-a□"$2"□!=□"-X"□]
do
□□□□□shift
□□□□□i□=□`expr□$i□+□1`
□□□□□echo□"message□part□$i□=□$1"□>>□FILE
done
```

4.2.7 DSM Event Reporting

DSM Event Reporting is a separate program package for error diagnosis in SNA networks. It lets error messages be combined from all systems connected to the network so that reactions to them can be triggered.

Two requirements have to be fulfilled in order for messages to be forwarded to DSM Event Reporting:

- The option *Connect to Event Reporting DSM-ER* must be set when Logging is launched (see Figure "Dialog box for setting start options").
- The forwarding of messages must be specified in the component assignment. All messages from the appropriate components will then be passed on to DSM Event Reporting (see Figure "Dialog box for modifying a component assignment").

See the "DSM Event Reporting, Monitoring Events in Networks" manual for more details on DSM Event Reporting.

4.2.8 TransView

TransView is a separate program package for system and network administration, as well as administration of applications in complex networks. The central component of TransView is the TransView Control Center (TV-CC) which bundles various administrative tools to form an all-in solution and lets users keep track of and control distributed resources in the network. Among other things, TV-CC is responsible for collecting, filtering and presenting all messages and events relating to the administered network objects at a central location. See the "TransView Control Center" manual for more details of TV-CC.

There are various ways of passing messages to TransView:

- You can set the *Write protocol for TransView* option when Logging is started (see Figure "Dialog box for setting start options"). All messages relating to the component that is to be passed on according to the component assignment are then passed to TransView (see Figure "Dialog box for modifying a component assignment").
- When certain events occur, you can call the *Call TransView* reaction from Event Management (see Figure "Dialog box for creating an event definition"). This is the standard procedure.

5 Evaluation of system device logs

5.1 Logging methods

SCSI device input/output errors are constantly logged in the background by the *errord*(8) daemon (see the section Reliant UNIX error messages (RM600) or (RM200/300/400) ...).

Error information is stored in compressed form in the log file */var/adm/error/errfile.current.Z*.

This file is recreated whenever the system is powered up. In this process, the name of the formerly current file is changed from */var/adm/error/errfile.current.Z* to */var/adm/error/errfile.previous.Z*, and the file that was previously named */var/adm/error/errfile.previous.Z* becomes */var/adm/error/errfile.<date>.Z*. The new file */var/adm/error/errfile.<date>.Z* thus always contains the date (and time) of the last power-up in its name.

A user interface is available for you to evaluate the error logs. It has been integrated in SYSADM and in the diagnosis menu (see the section Error and status messages).

The following section describes how error logs are evaluated with the interface that is integrated in SYSADM. Handling of SYSADM is described in the manual "System Administration and Hardware Configuration Using the SYSADM Interface".

5.2 User interface

You call the user interface to evaluate SCSI error logs by means of the *logs — SCSI devices* menu item.

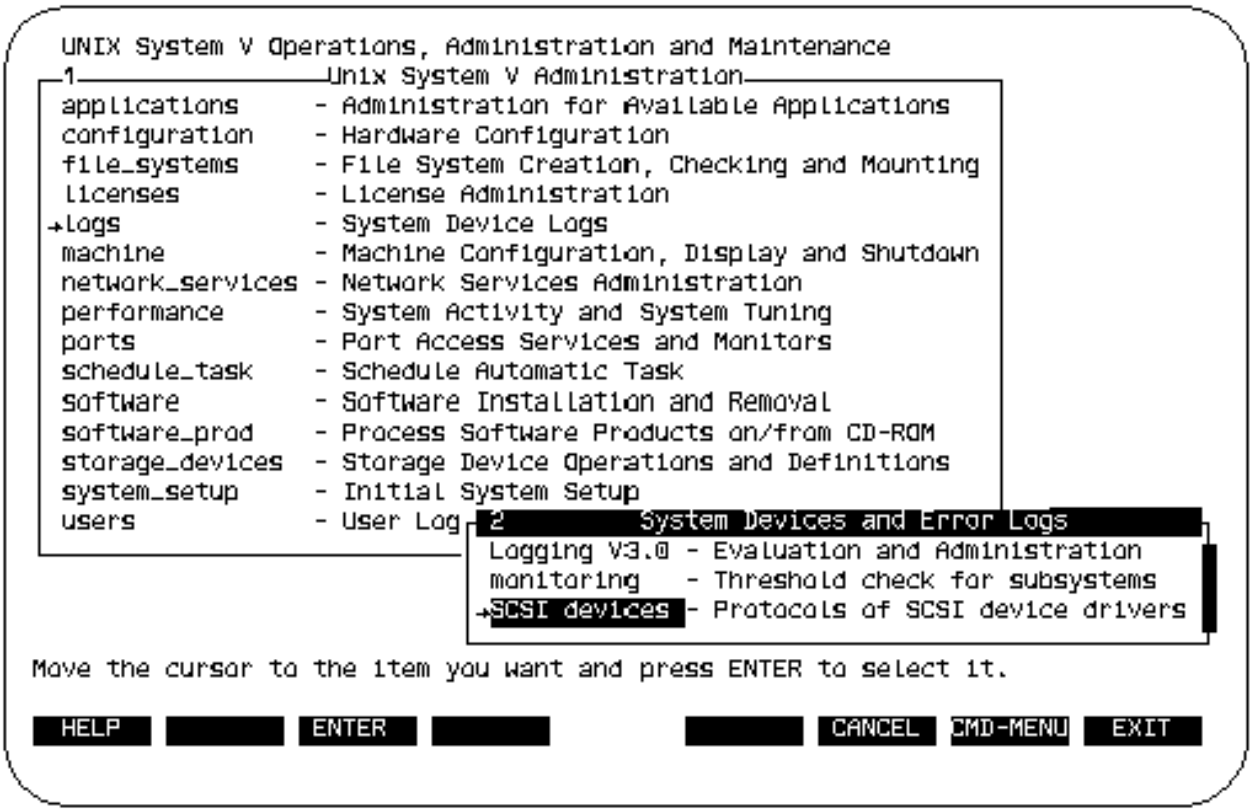


Figure 17: Calling the user interface to evaluate SCSI error logs

Calling the SCSI device menu opens a window in which you can choose whether you want to evaluate individual error messages or entire log files. You can also delete log files there.

5.2.1 Evaluating individual error messages

The advantage of this method of evaluation is that you get quick access to specific error messages. You don't have to find the required log file first and then look for the messages in it. For evaluation of entire log files see the section Evaluating log files

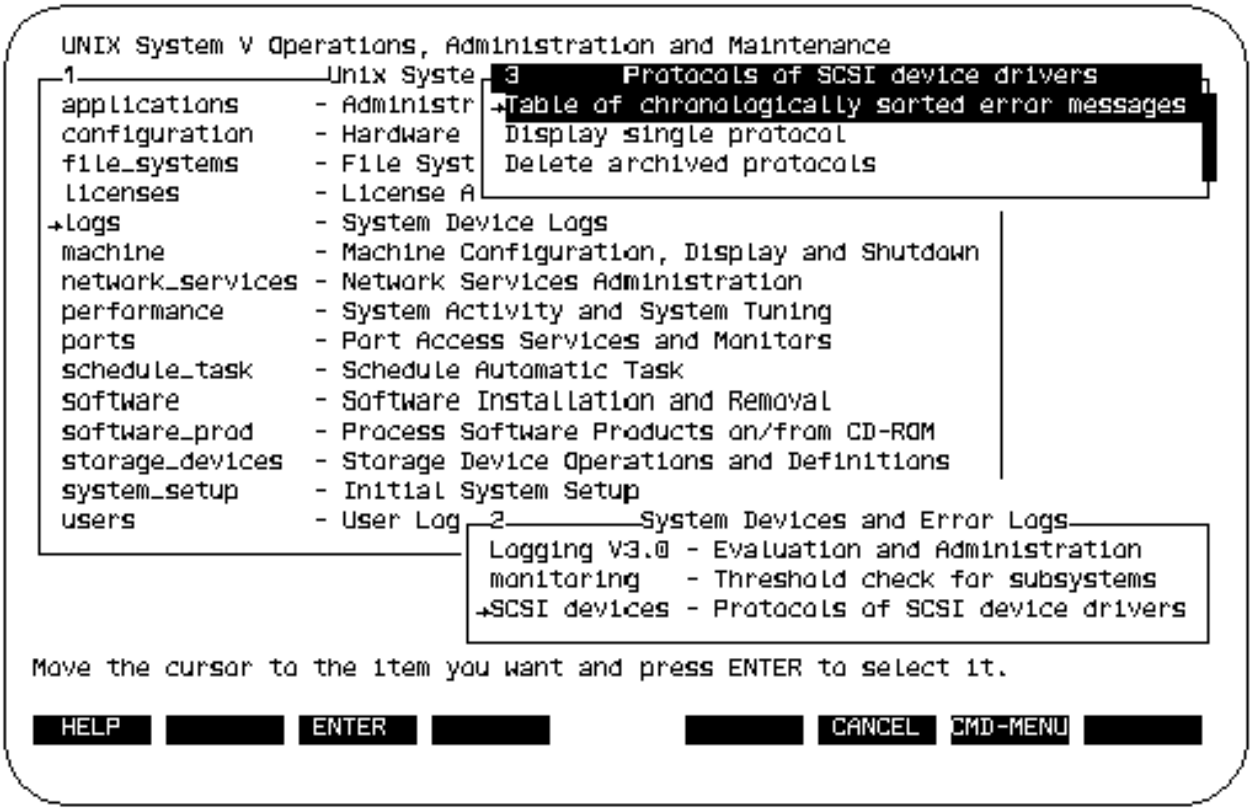


Figure 18: Evaluating individual error messages

Selecting the *Table of chronologically sorted error messages* menu item displays a list of all error messages in reverse chronological order (the first entry is the most recent).

This list does not include any messages that are generated whenever the system is powered up and indicate which kernel release has been booted (so-called UTS messages). An example of a UTS message is given as the first entry in Figure "Displaying a selected log file (ENTER softkey)".

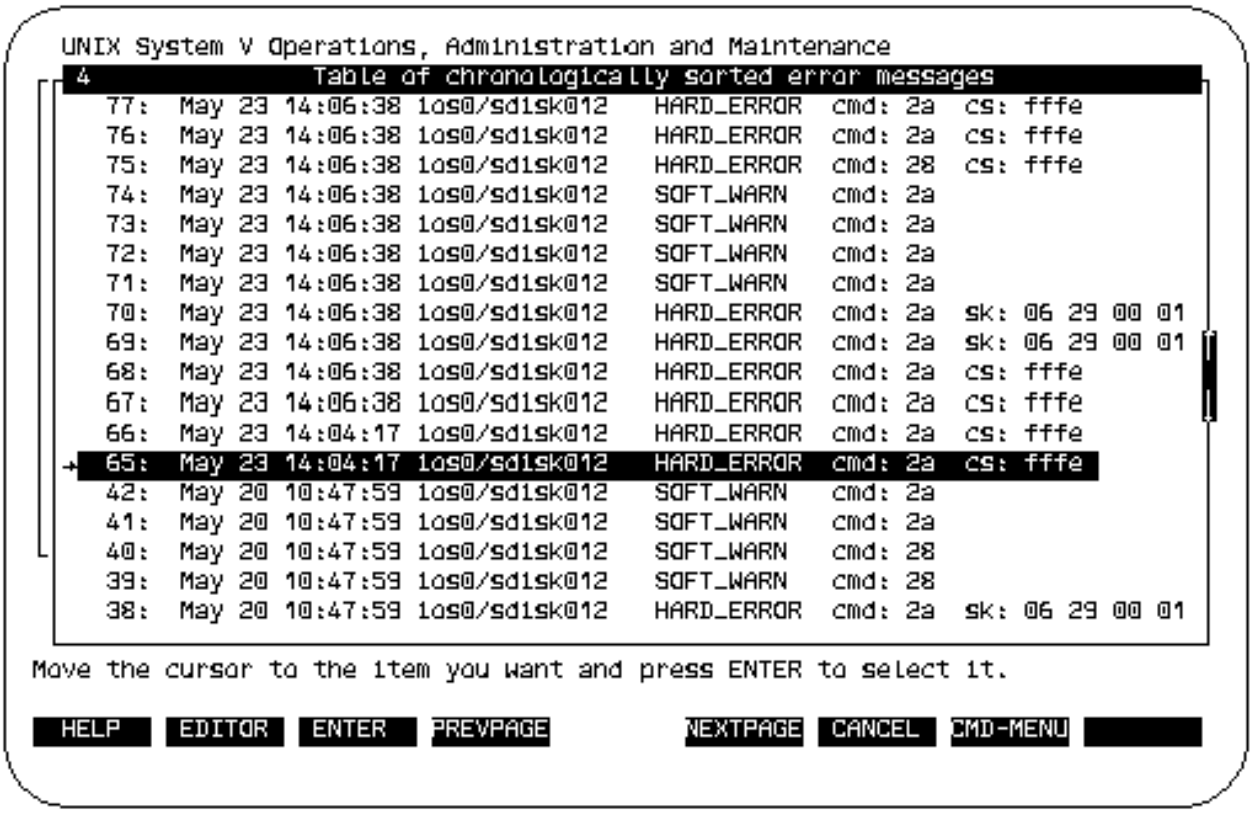


Figure 19: List of error messages

Pressing the EDITOR softkey calls the editor that is defined in the *EDITOR* environment variable. All messages, including UTS messages, are displayed in detail and the cursor is positioned on the selected message. Here you can use the editor's facilities to search through the output.

Pressing the ENTER softkey displays the selected error message in detail together with the next five.

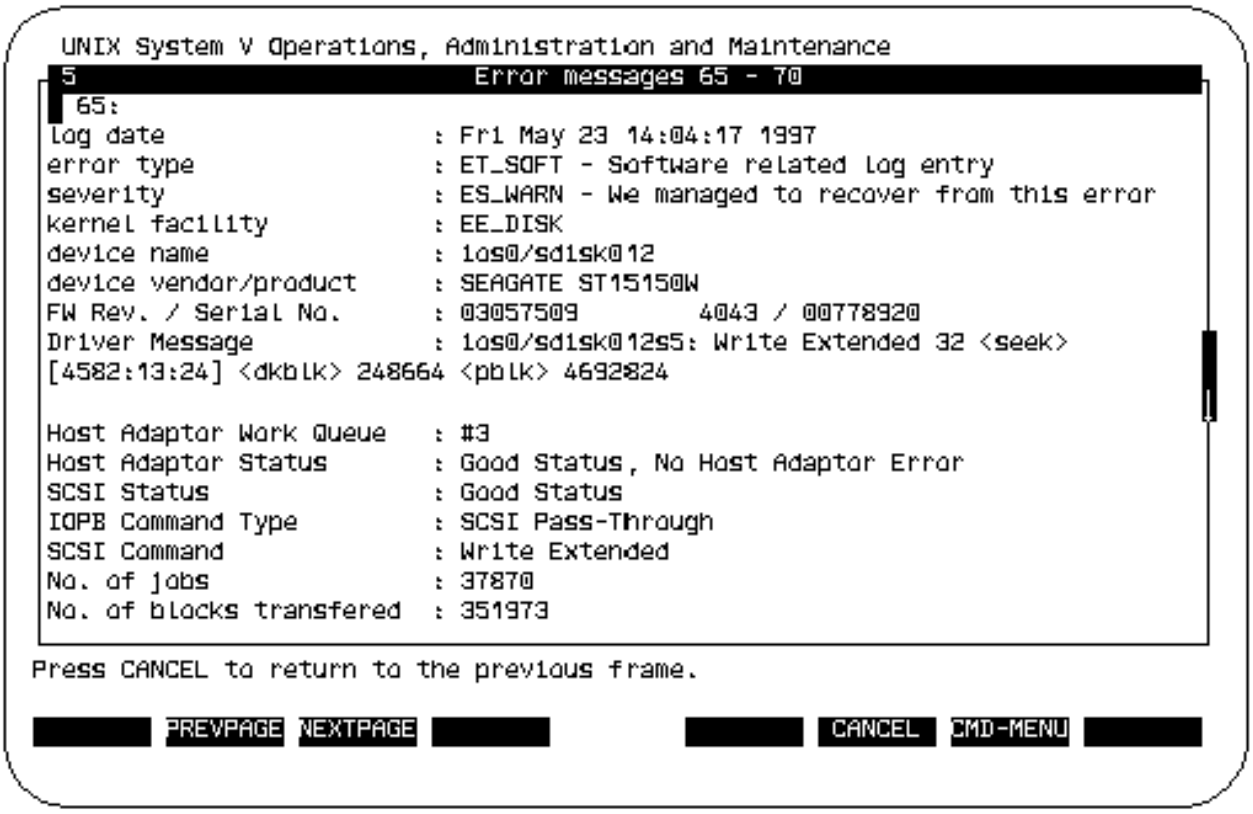


Figure 20: Displaying a selected error message (ENTER softkey)

5.2.2 Evaluating log files

Selecting the *Display single protocol* menu item brings up a list of all log files named *errfile*.Z* in reverse chronological order. Files for which a hyphen is shown instead of a size only contain a UTS message which is generated whenever the system is powered up and indicates which kernel release was booted.

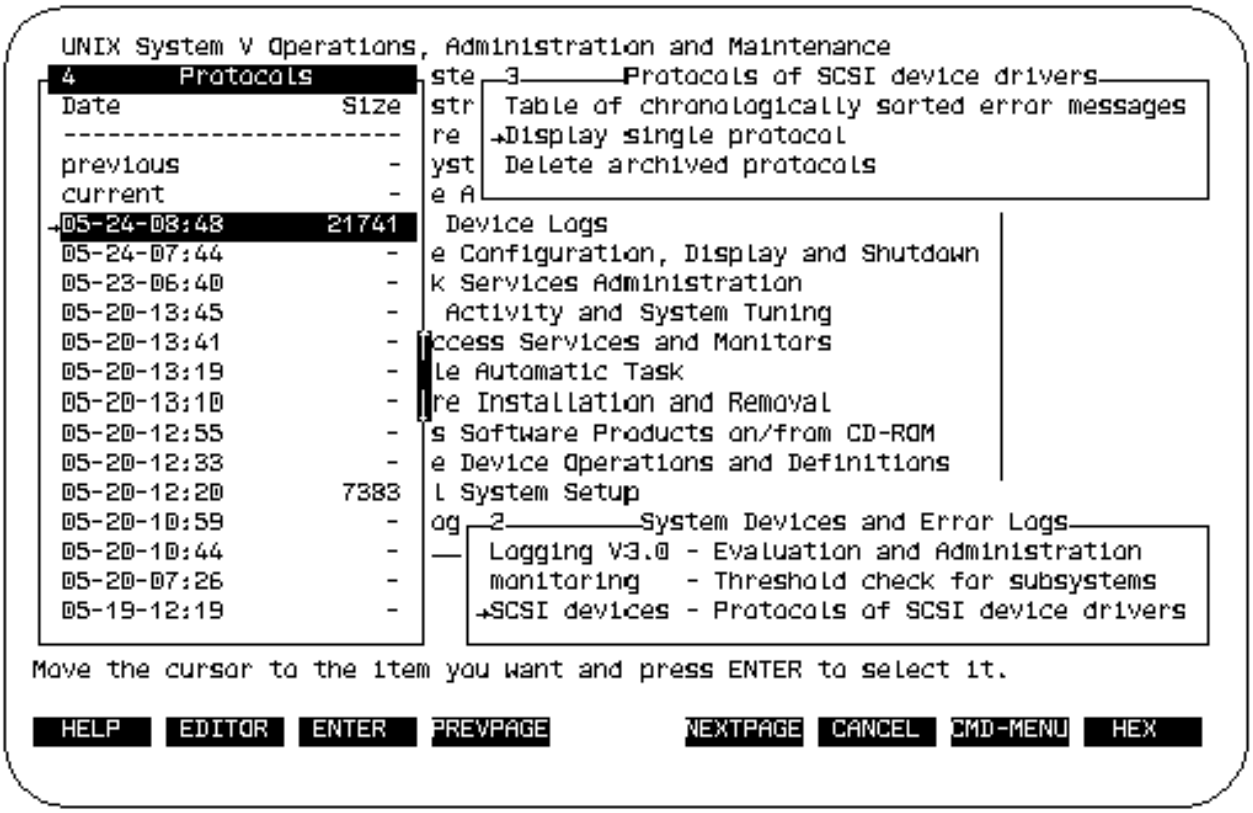


Figure 21: List of log files

Pressing the EDITOR softkey calls the editor that is defined in the *EDITOR* environment variable. The selected log file is displayed in detail. Here you can use the editor's facilities to search through the output.

Pressing the HEX softkey calls the editor that is defined in the *EDITOR* environment variable. The content of the selected log file is displayed in hexadecimal notation.

Pressing the ENTER softkey displays the selected log file in detail.

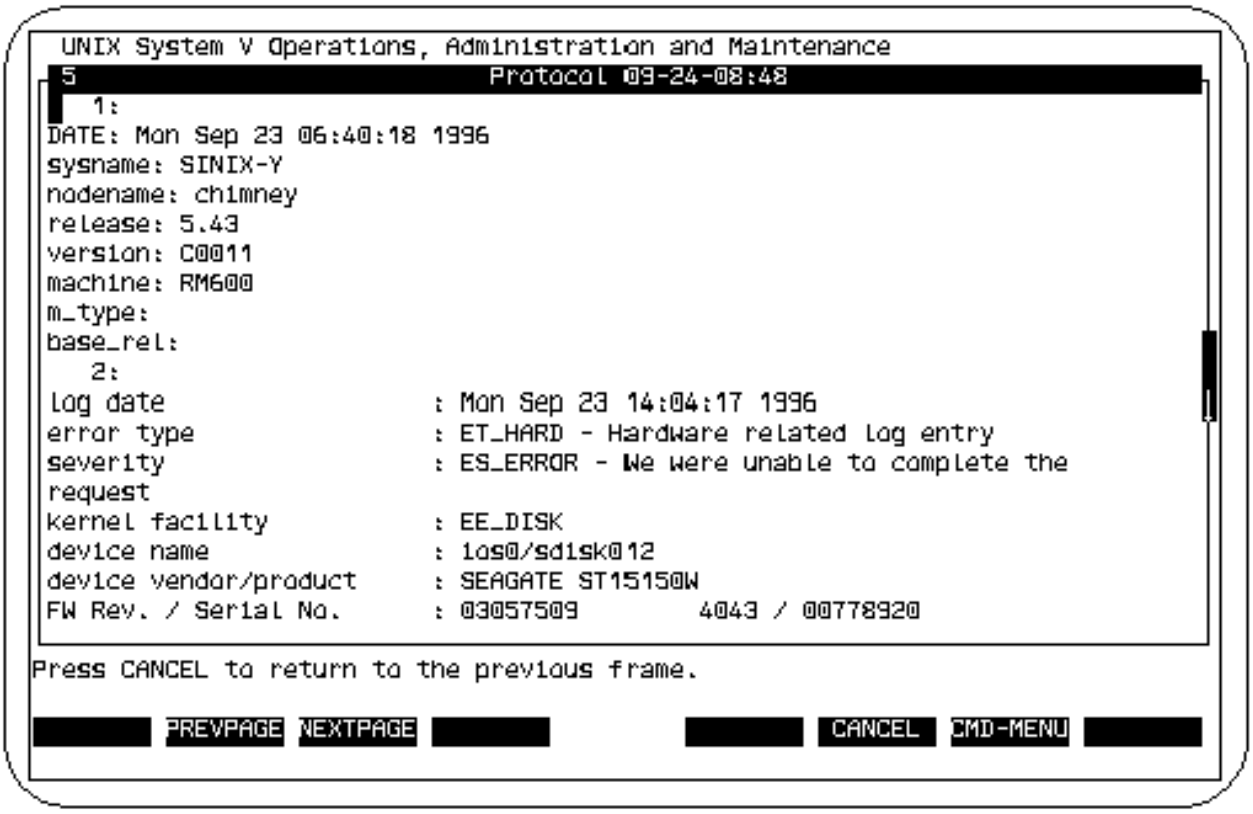


Figure 22: Displaying a selected log file (ENTER softkey)

5.2.3 Deleting log files

Selecting the *Delete archived protocols* menu item brings up a list of all archived log files apart from *current* and *previous*.

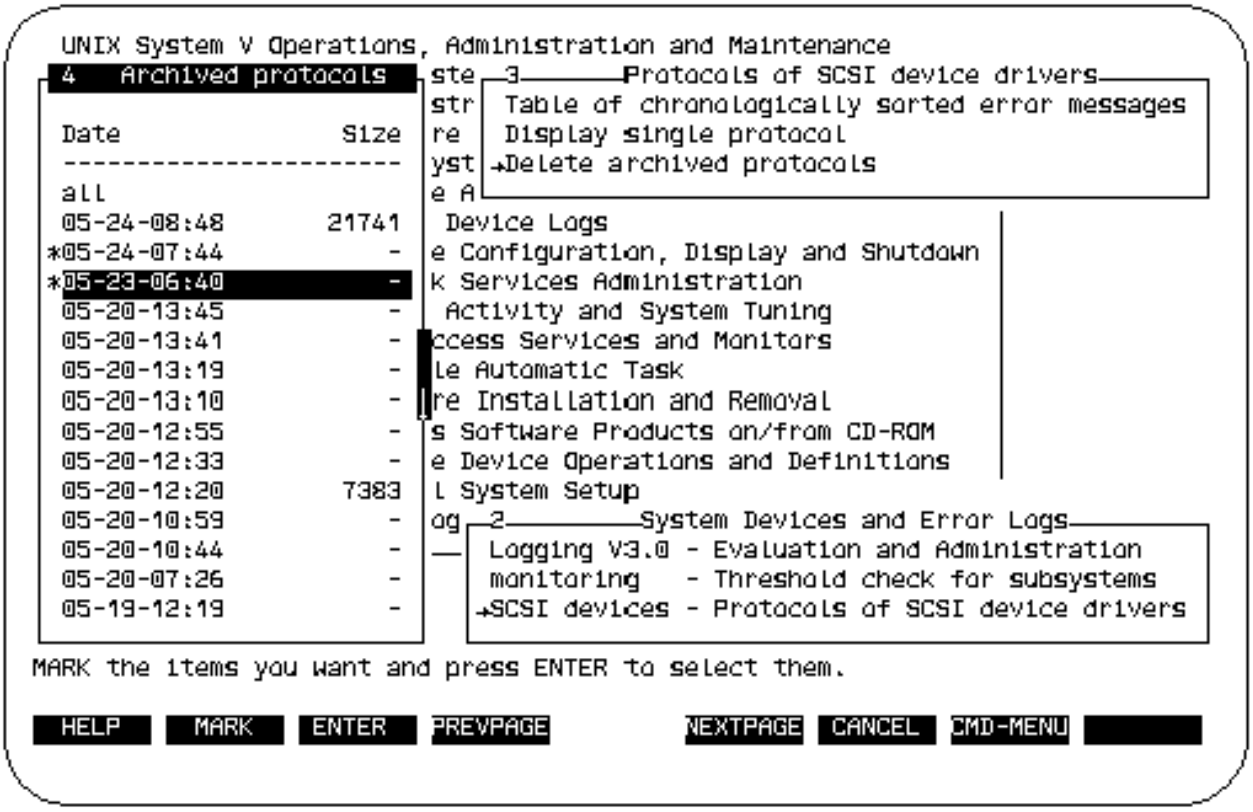


Figure 23: Deleting log files

To delete log files you first have to mark them with the MARK key and then confirm this selection with the ENTER softkey. This opens a window in which you have to confirm deletion again with ENTER as a precaution. Then the files are deleted.

6 Hardware diagnostics using VConfig and Config

You can use VConfig to obtain information about the status of hardware components, both on a local system and within an administration domain, and on RM600 systems you can also initiate changes to that status . You launch VConfig from the graphical user interface WebSysAdmin (See the manual "System Administration within a Domain".).

On local systems, besides the hardware diagnostic facilities of VConfig, you can also still use some features of Config under the character-oriented user interface SYSADM (See the manual "System Administration and Hardware Configuration Using the SYSADM Interface".).

VConfig is not available on computers of type RM200-xxx and RM400-xxx. Here you can perform hardware diagnostics functions using Config under the graphical user interface SINIX/windows (See the manual "Hardware Configuration with Config under SINIX/windows".) or the character-oriented user interface SYSADM.

In order to understand this chapter you should be familiar with the LAR concept (see the chapter The LAR concept).

On RM600 systems you can obtain information about whether

- boards, components of boards (such as controllers) and SCSI devices are faulty, defective or deactivated.
- BBUs, power units or fans are defective or there are problems with overheating (the latter only if you are using Config under SYSADM).

On RM600 systems you can also use VConfig for the following:

- manually setting the status of boards, components of boards and SCSI devices to *defect* using the function *Deactivate*, for example because you suspect a fault.
- clearing the status *error*, *defect* or *deactivated* for boards, components of boards and SCSI devices using the function *Activate*. There is, however, no point in doing this unless you are quite sure that the fault has been cleared, or if the status was set to *defect* manually.

On RM200/300/400 systems you can obtain information about whether

- CPUs and main memory modules are defective or deactivated.
- SCSI devices are defective or deactivated.
- BBUs, power units or fans are defective, the primary power supply has failed, or there are problems with overheating (the latter only if you are using Config under SYSADM or SINIX/windows).

You cannot manually set or clear flags on hardware components of RM200/300/400 systems using the user interfaces, but you can set and clear them manually for these systems using shell commands (see the section LAR on computers of type RM200/RM300/RM400).

6.1 General information

VConfig

In the Hierarchy Browser or the Object Table you will find the following information regarding the status of the individual computers and hardware components:

The computer is depicted in the Hierarchy Browser on a gray field
: The computer is not available at present, for example because it is powered down.

Yellow dot
: The object is faulty. It may remain in operation for the time being but should be replaced as soon as convenient.

Red dot
: The object is defective and must be replaced.

Red and black dot
: The object is defective and must be replaced. It will be deactivated the next time the system is booted.

Black dot
: The object is deactivated.

Black and yellow dot
: The object is still deactivated. It will be reactivated the next time the system is booted, is however faulty.

Black and green dot
: The object is still deactivated. It will be reactivated the next time the system is booted and is fault-free.

Faulty, defective or deactivated hardware components are also displayed on a colored field in the tab *Hardware Presentation*:

Light gree
: The object is fault-free.

Dark green
: The object contains deactivated hardware. The deactivated hardware is not faulty and will be reactivated at the next system startup.

Light yellow
: The object contains faulty hardware. The faulty hardware can remain in service for the moment but should be replaced at an appropriate time.

Dark yellow
: The object contains deactivated hardware. The deactivated hardware is faulty but will be reactivated at the next system startup.

Light red
: The object contains defective hardware. The defective hardware must be replaced.

Dark red
: The object contains defective hardware. The defective hardware must be replaced and will be

deactivated at the next system startup.

Dark grey
: The object contains deactivated hardware.

Config under SINIX/windows

If you have a color monitor the icon view shows faulty, defective or deactivated components on a red field. Monochrome monitors use a different gray shade instead (See the chapters "Setting the display" and "Defective and deactivated devices" in the manual "Hardware Configuration with Config under SINIX/windows").

Message windows in VConfig and Config

A message window will inform you if faults arise while the system is in operation. If you are using Config, you will then have to perform the proposed analysis function; the only other option you have at this point is to terminate the session.

6.2 Detailed information

Faulty boards, components of boards and SCSI devices

Under VConfig, faulty hardware is flagged in dialog boxes with *error* in the field *Status LAR*.

Under Config, faulty hardware is flagged in attribute windows with *Active* in the field *Status* and with *error* in the field *Status Info*.

Defective boards, components of boards and SCSI devices

Under VConfig, defective hardware is flagged in dialog boxes with *defect* in the field *Status LAR*. If the defective hardware is deactivated at the next boot, it is then flagged with *defect-deactive*.

Under Config, defective hardware is flagged in attribute windows with *active* in the field *Status*, and in the field *Status Info* with *defect* (if the flag was set automatically) or *deactive* (if it was set manually).

Deactivated boards, components of boards and SCSI devices

Under VConfig, deactivated hardware is flagged in dialog boxes with *deactive* in the field *Status LAR*. If the deactivated hardware is reactivated at the next boot, it is then flagged with *deactive-ok* (active and fault-free) or *deactive-error* (active but still faulty).

Under Config, deactivated hardware is flagged in attribute windows with *Inactive* in the field *Status* and in the field *Status Info* with *defect* (if the flag was set automatically) or *deactive* (if it was set manually).

Defective external BBUs

If the system is equipped with an external BBU, defective DC/DC transformers are indicated as follows:

- under VConfig they are shown on a red field in the *Hardware Presentation* of the BBU cabinet.
- under Config they are flagged with *defect* in the field *Status of no.<n>* of the attribute window for the BBU cabinet.

Defective internal BBUs

If a cabinet is equipped with an internal BBU, a defective BBU is indicated as follows:

- under VConfig it is shown on a red field in the *Hardware Presentation* of the relevant cabinet.
- under Config it is flagged with *defect* in the field *BBU Monitoring* of the *Cabinet Information* window for the relevant cabinet.

Defective power units

Defective power units are indicated as follows:

- under VConfig they are shown on a red field in the *Hardware Presentation* of the relevant cabinet.
- under Config they are flagged with *defect* in the field *Power unit <n>:* of the *Cabinet Information* window for the relevant cabinet.

Defective fans

Defective fans are indicated as follows:

- under VConfig they are shown in the *Hardware Presentation* of the relevant cabinet on a red (total fan failure) or yellow field (redundant fan failure).
- under Config they are flagged with *Multi failure* (total fan failure) or *Single failure* (redundant fan failure) in the field *Fan drawer <n>:* of the *Cabinet information* window for the relevant cabinet.

Temperature problems

Problems with the temperature are displayed only under Config; they are flagged with *Temperature failure* in the field *Fan drawer <n>:* of the *Cabinet information* window for the relevant cabinet.

Primary power failure (RM200/300/400)

Failure of the primary power supply is displayed only under Config: it is flagged with *defect* in the field *Power unit <n>:* of the *Cabinet information* window for the relevant cabinet.

7 Provision of error information in repair cases

When a hardware component has been sent in for repair, the information in the system can be provided to analyze the problem.

With this method, all information pertaining to a defective hardware component is gathered (partial dump) and written to a storage medium. This storage medium should be sent to Service together with the hardware component to help analyze the problem.

The following user interface is available in the Diagnostics menu to gather error information and write it to a storage medium(see also the section Dump handling):

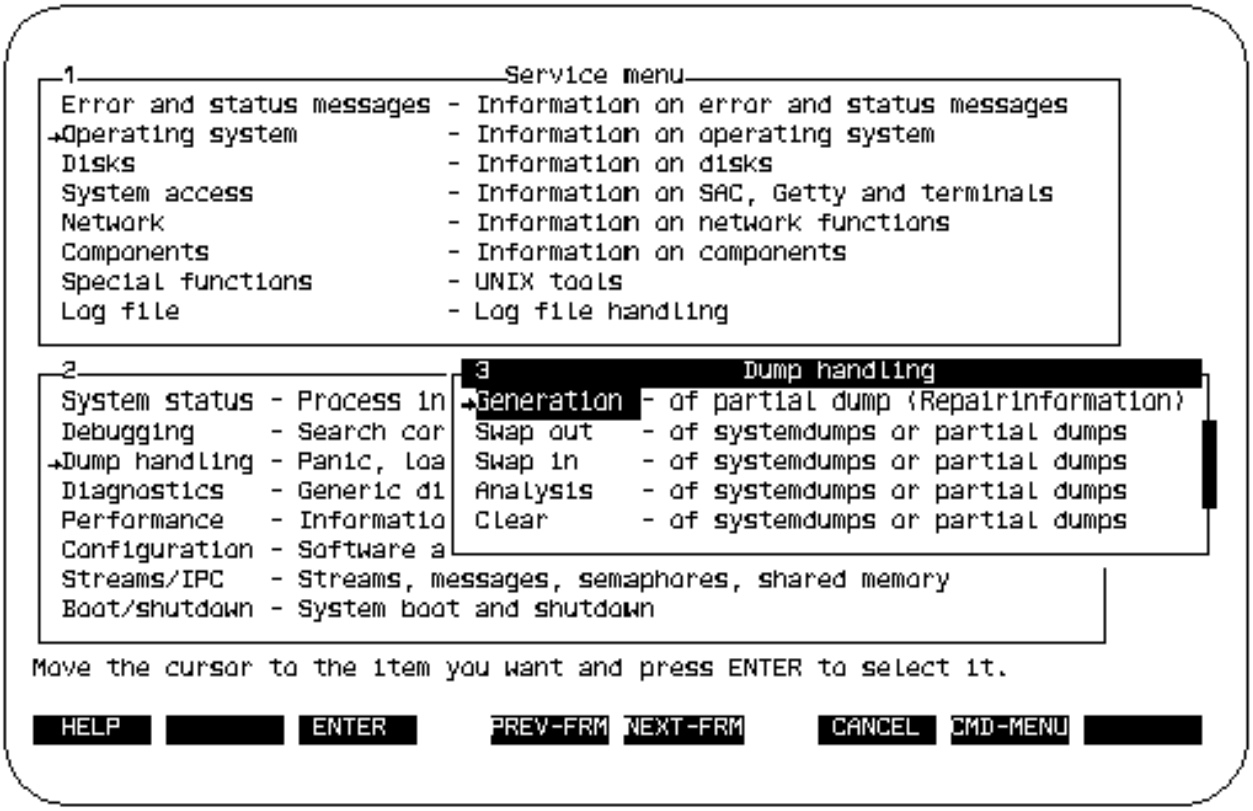


Figure 24: User interface for dump handling in the Diagnostics menu

The relevant error information is gathered under the *Generation* menu item.

So that the system can gather the relevant error information, a window is opened when you call this menu item and you have to enter the following details at least:
Name of the defective component
Name of the originator of the report
Name of the customer
Description of the error (*Comment* input field)

All other input fields on this form are optional and serve to provide more details of the component, the order and the cause of the error.

You can write the error information that has been gathered to a storage medium by calling *Swap out* from the menu.

One volume of the storage medium may not be sufficient; you may need to have one or more continuation volumes ready when you swap out the data.

Calling this menu item opens a window which contains a list of all dumps on your system.

When you select one of these dumps, another window opens so that you can check the details you recorded for this dump under *Generation* and change them if necessary.

Quitting this window with the SAVE softkey opens another window in which you have to specify the output device and can define additional swap-out files if necessary. Pressing the SAVE softkey again starts the output.

8 Diagnosis menu for Reliant UNIX systems

The diagnosis menu is available to give you an in-depth picture of a system that you want to diagnose. This includes the existing configuration of the system (configuration and device files), online observation of system activities (to locate bottlenecks and tune the system) and a systematic analysis of log files.

The facilities provided by the diagnosis menu are of a passive nature only, i.e. it is not possible to make changes to the system (data security) or read data that you are not unauthorized to access (data protection).

This is a menu-driven user interface that has been implemented with the aid of FMLI (Form and Menu Language Interpreter).

8.1 Guide to this chapter

This chapter presents the menu-driven system diagnostics in Reliant UNIX. It only shows the diagnostic options and not how to remedy errors.

Depending on the hardware platform and/or the configuration you are using, some of the menu items may not be present.

You can display the machine name and current software release using the command *uname -a* or from the diagnosis menu via *Operating system — System status — Machine information*.

The diagnosis menu is a menu system that draws upon shell commands to carry out the action that you select. This is why the commands underlying the menu items are shown in tables as well.

In some cases, an item on the diagnosis menu displays the contents of a file or directory. In these cases, the file or directory name is shown.

8.2 Special softkeys

In addition to the standard softkeys that are available under FMLI, the diagnosis menu has two special softkeys on the lower levels of the menu:

MANUAL

This softkey lets you view the online manual page for the command that corresponds to a menu item.

PROTO

This softkey lets you write the output shown in the text window to the log file named */home/diagnose/proto*. The date, time and menu item that initiated the output are logged for each output. This log file can be displayed, printed and cleared via *Log file — Output*.

8.3 Calling and closing the diagnosis menu

Log in to the system as the *diagnose* user.

You are asked for your password. The diagnosis menu user interface appears when you have entered it.

A password must have been assigned for the *diagnose* user before the diagnosis menu can be called the first time.

You close the diagnosis menu with the *exit* command in the command menu which you open with the CMD-MENU softkey. At the same time, this logs you off the system.

8.4 Error and status messages

This menu item enables you to:

- Evaluate error messages and status messages from SCSI device drivers
- Check on the status of the hardware environment
- Evaluate log entries from the log system Logging
- Display system messages from the most recent boot procedure
- Display the configuration file and the message files belonging to the daemon *syslogd*
- Display Streams error messages
- Display the error descriptions corresponding to Reliant UNIX errors
- Evaluate SPbus error logs
- Configure the directory for SPbus error logs

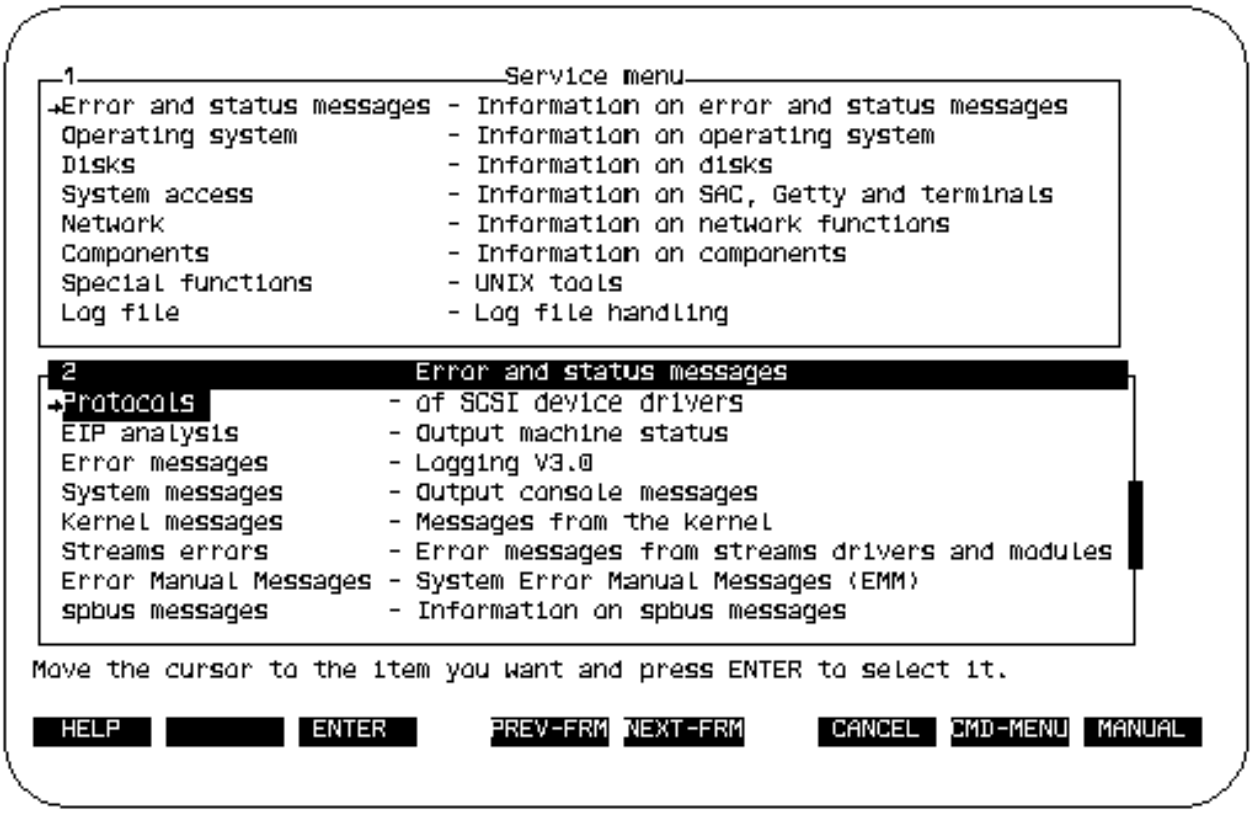


Figure 25: Diagnosis menu: Error and status messages

8.4.1 Protocols

You can use this menu item to evaluate error messages and status messages from SCSI device drivers:

Table of chronologically sorted error messages
: Display error messages from the */var/adm/error/errfile.** error messages files

Display single protocol
: Display single error log

Delete archived protocols
: Delete error logs using the *rm errfile.** command

8.4.2 EIP analysis

You can use this menu item to check on the status of the hardware environment using the *ups -s* command.

8.4.3 Error messages

You can evaluate log entries of Logging from this submenu.

8.4.4 System messages

You can use this menu item to display the system messages from the most recent boot procedure.

8.4.5 Kernel messages

You can use this menu item to display the configuration file belonging to the *syslogd* daemon and messages from the operating system kernel:

Configuration file
: Display */etc/syslog.conf* configuration file

Messages
: Display */var/adm/log/messages* file

Debugmsgs
: Display */var/adm/log/debugmsgs* file

The menu item *Debugmsgs* will be displayed only if the */var/adm/log/debugmsgs* file is listed in the */etc/syslog.conf* configuration file.

8.4.6 Streams errors

You can use this menu item to evaluate error messages from Streams drivers and Streams modules:

Trace infos
: Display trace information using the *strace* command

Trace infos outputs the Streams monitor messages for all drivers and modules. These messages come from the *log*(7) Streams log driver. Once strace has been started, it stays active until you terminate it by pressing the [CTRL][C] and CR keys.

Error log
: Display */var/adm/streams* directory

8.4.7 Error Manual Messages

You can use this menu item to display the error descriptions corresponding to Reliant UNIX errors:

Catalog
: Select catalog definition using the *getemm <catalog>* command

Error number
: Select error number using the *getemm <number>* command

8.4.8 Spbus messages

You can use this menu item to display the error logs relating to SPbus errors, and to specify which directory is to be used for such error logs and the maximum number of error logs which may be present in this directory:

spbus error dumps
: Display error logs

configuration
: Establish directory and number of logs

8.5 Operating system

This menu item gives you information on the following:

- Process and machine details and system settings
- System performance
- Software and hardware configuration
- Streams and interprocess communication
- Boot, init and shutdown processes

You can also do the following:

- Search for a core dump in a directory and trace a selected process
- Analyze and swap out core dumps, and swap them back in
- Analyze, generate, transfer and clear diagnostic files
- Run a consistency check on the firmware

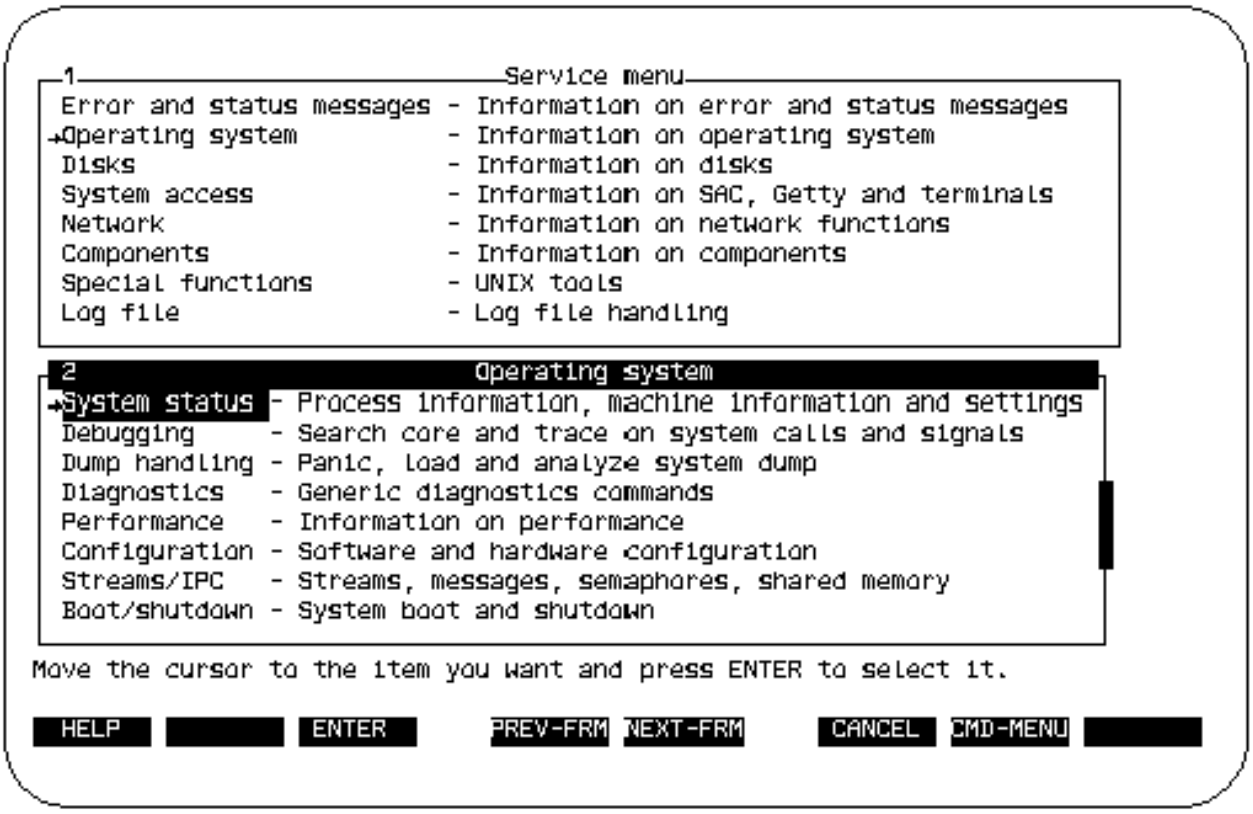


Figure 26: Diagnosis menu: Operating system

8.5.1 System status

This menu item lets you look up process and machine details and system settings:

Run level
: Display run level of the system using the *who -r* command

Active user IDs
: Display all active user IDs in the system using the *who -a* command

Process information
: Display information on user processes running using the *whodo -l* command

Started processes
: Display processes started by users using the *whodo* command

Interprocess communication
: Display active facilities for interprocess communication using the *ipcs -a* command

Date
: Display date and time using the *date* command

Machine information
: Display machine name and release using the *uname -a* command

Processes in progress
: Display all processes running on the system using the *ps -ef* command

Time zone
: Display time zone that has been set up on the machine from the */etc/TIMEZONE* file

8.5.2 Debugging

This menu item lets you search for a core dump in a directory and trace a process:

Core Find a core dump in a directory using the *find <directory> -name core -print* command

Trace Trace system calls and signals using the *truss* command

You terminate the trace by entering [CTRL][C].

8.5.3 Dump handling

This menu item lets you generate a partial core dump, analyze a system core dump or partial core dump, clear it and swap it out to tape or diskette and swap it back in:

Generation
Generate a partial dump (repair information) using the *repair* command

Swap out
Swap out system dump or partial dump using the *cpio -ocvC8192* command

Swap in
Swap in system dump or partial dump using the *cpio -imcdvC8192* command

Analysis
Analyze system dump or partial dump using the *crash* command

Clear Clear system dump or partial dump using the *rm -rf* command

The directory which is used for the dump is the one laid down by the system variable *DUMPDIR* in the */etc/default/dumpsave* file (the default is */var/crash*). This directory must have write permissions so that you can swap dumps both in and out. The variable *EXTDUMP* must be set up in the */etc/default/dumpsave* file before you can swap in a dump. The files to be swapped out (*dump.x* and *unix.x*) must possess read permissions.

8.5.4 Diagnostics

This menu item gives you information on the *dptg*, *rtty* and *SIsih* drivers. In the event of errors, this information can help you find the cause.

This menu item should only be called after consultation with the Development department.

8.5.5 Performance

This menu item gives you information on system activities, system settings, swapping/paging and the state of the multiprocessor architecture (RM600 only).

System utilization — Processes — Process1
Display running processes using the *ps -ef* command

System utilization — Processes — Process2
Display frequently requested processes using the */usr/ucb/ps -a* command

System utilization — Users — Process information
Output information on users' running processes using the *whodo -l* command

System utilization — Users — Started processes
Display all processes started by users using the *whodo* command

System utilization — System activities — File access
Output use of system routines for file access using the *sar -a* command

System utilization — System activities — Buffers
Output buffer activities using the *sar -b* command

System utilization — System activities — System calls

Output counters for system calls using the *sar -c* command

System utilization — System activities — Block oriented devices

Output activities of block devices using the *sar -d* command

System utilization — System activities — Page swap out
Output page swap out activities using the *sar -g* command

System utilization — System activities — System core memories
Output system core memory assignments using the *sar -k* command

System utilization — System activities — Messages/semaphores
Output message and semaphore activities using the *sar -m* command

System utilization — System activities — Page swap in
Output page swap in activities using the *sar -p* command

System utilization — System activities — Queue
Output utilization and length of queue using the *sar -q* command

System utilization — System activities — Memory utilization
Output unused memory pages and disk blocks using the *sar -r* command

System utilization — System activities — CPU
Output CPU utilization using the *sar -u* command

System utilization — System activities — System tables
Output status of system tables using the *sar -v* command

System utilization — System activities — Swapping
Output swapping activities using the *sar -w* command

System utilization — System activities — File system
Output file system operations performed using the *sar -x* command

System utilization — System activities — TTY devices
Output TTY device activities using the *sar -y* command

System utilization — Virtual memory
Output virtual memory statistics using the *vmstat* command

System utilization — Forks
Output fork statistics using the *vmstat -f* command

System utilization — Directories
Output large directories using the *find / -type d -size +40 -print* command

Tunables — Parameter setting
Display system settings for adjustable parameters from the */etc/conf/cf.d/stune* file

Tunables — Default setting
Display default settings for adjustable parameters from the */etc/conf/cf.d/mtune* file

Swapping/paging — Swap areas
Display disk swap areas using the *swap -l* command

Swapping/paging — Swap blocks
Display swap blocks using the *swap -s* command

Swapping/paging — Swapping
Display swapping activities using the *sar -w* command

Swapping/paging — Paging
Display paging activities using the *sar -g -w* command

Swapping/paging — Since boot
Display swapping and paging activities since last boot using the *vmstat -s* command

Multiprocessor — Number
Display number of CPUs configured using the *mpcntl -N* command

Multiprocessor — Process info
Display process and CPU information using the *mpcntl -f -p* command

Multiprocessor — CPU info
Display information on active CPUs using the *mpcntl -l* command

Multiprocessor — Statistics
Display multiprocessor statistics using the *mpstat* command

If you have not set the polling interval in the statistics form, or have set it to a high figure, then you can terminate output or the menu item by entering [CTRL][C].

8.5.6 Configuration

This menu item gives you information on the software, hardware and controller configuration. You can also run a consistency check on the firmware:

SW configuration — System definitions
: Display system definitions using the *sysdef* command

SW configuration — Limit values
: Display system limits from the */usr/include/limits.h* file

SW configuration — System components
: Display system components using the *idinstall - g -s* command

SW configuration — Space.c
: Display *Space.c* component using the *idinstall - g -p* command

SW configuration — Stubs.c
: Display *Stubs.c* component using the *idinstall - g -t* command

SW configuration — Node
: Display *Node* component (device file) using the *idinstall - g -n* command

SW configuration — File type configuration
: Display *Mfsys* component using the *idinstall -g -c* command

SW configuration — Active file types
: Display *Sfsys* component using the *idinstall -g -l* command

SW configuration — Device drivers/SW modules
: Display device drivers and configurable software modules from the */etc/conf/cf.d/mdevice* file

SW configuration — System component files
: Display system component files directory from the */etc/conf/sdevice.d* file

SW configuration — Nodes — Nodes directory
: Display */dev* node directories

SW configuration — Nodes — Config directory
: Display */etc/conf/node.d* configuration file directory using the*ls -C* ommand

SW configuration — Nodes — Config files
: Display configuration files for nodes from the */etc/conf/node.d/<file_name>* file

SW configuration — Nodes — Normal files
: Display normal files in */dev* using the *find /dev -type f -print* command

HW configuration
: Display hardware configuration using the *autoconf -l -v* command

Controller
: Display system configuration of controllers using the *showconf* command

Firmware
: Check whether controllers and EIPs have the correct firmware revision levels using the */etc/init.d/fw_check* command.

8.5.7 Streams/IPC

This menu item gives you information on interprocess communication and the Streams configuration:

Interprocess communication
: Display IPC status using the *ipcs* command

Streams configuration — Streams modules
: Display Streams modules to be pushed automatically using the *autopush -g -M* command

Streams configuration — Directory
: Display */etc/ap* autopush configuration file directory

Streams configuration — Files
: Display */etc/ap/<file_name>.ap* autopush configuration files

Streams configuration — TCP/IP
: Display Streams configuration for Streams TCP/IP from the */etc/ntrcf* file

8.5.8 Boot/shutdown

This menu gives you information on the boot, init and shutdown processes:

Boot — Configuration file
: Display */etc/default/boot* configuration file

Boot — Directory
: Display */stand* kernel file system

Boot — Checksum
: Display checksum of */unix* and */stand/unix* using the *sum -r* command

Boot — Boot parameters
: Display boot flags, boot string and dump string using the *bootflags* command

Init — Init script
: Display */etc/inittab* manuscript file

Init — Script1
: Display */sbin/brc* system initialization process

Init — Script2
: Display */sbin/bcheckrc* system initialization process

Init — Directory1
: List *init.d* scripts from the */etc/init.d* directory

Init — Directory2
: List *rc* scripts from the */etc/rcn.d* directory

Init — Rc script
: Display */etc/rc<n>* scripts

Init — Run level
: Display run level of the system using the *who -r* command

Shutdown
: Display */sbin/shutdown* script to shut down the system

8.6 Disks

This menu item gives you information on the following:

- Configuration and state of disks
- Mounted file systems

You can also check file systems.

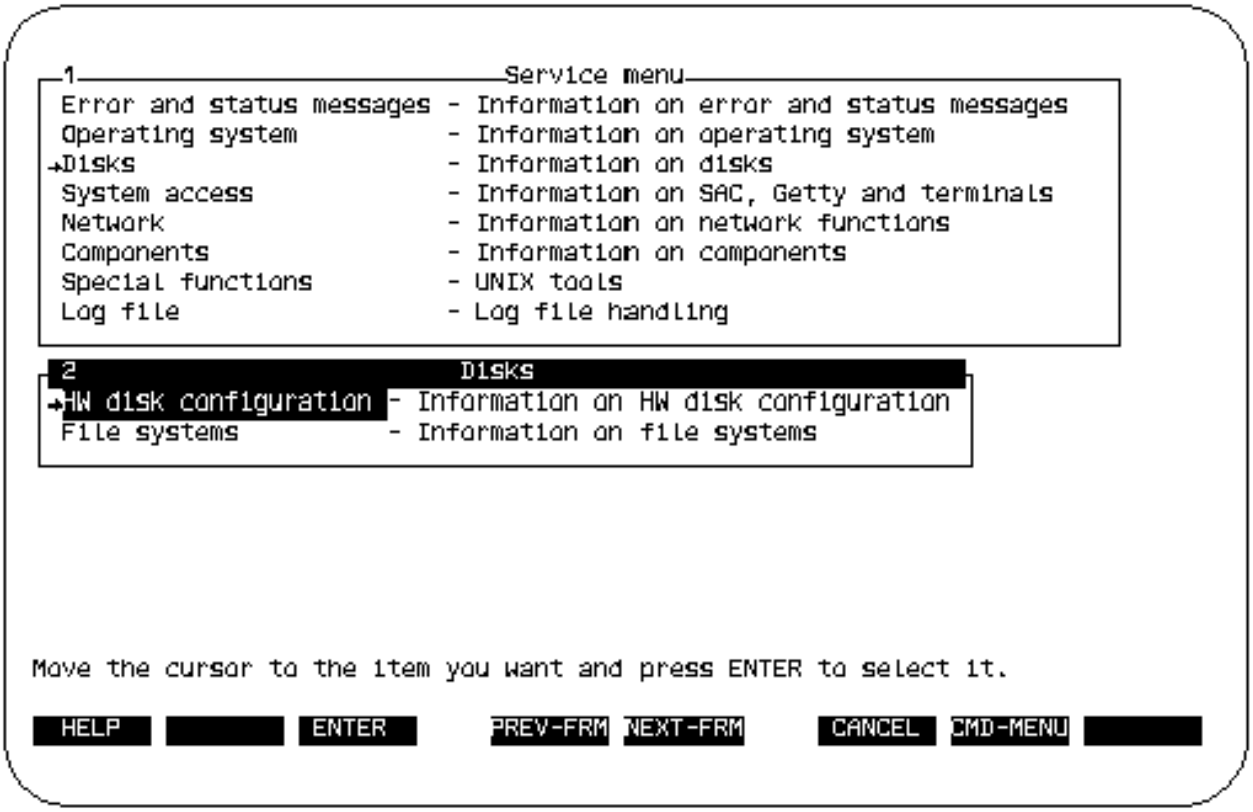


Figure 27: Diagnosis menu: Disks

8.6.1 HW disk configuration

This menu item gives you information on configuration, state and utilization of disks:

HW configuration
: Display hardware component information using the *autoconf -l -v* command

Disk partitioning
: Display disk partitioning using the *dkpart -l* command

Disk space usage
: Display map of disk space usage using the *dkmap* command

Virtual disk partitioning
: Display virtual disks currently configured using the *dkconfig -alv* command

Mirror disk status
: Display status of mirror virtual disks using the *dkmirror -lv* command

Disk utilization — Block oriented devices
: Display disk transfer statistics using the *dkstat -a* command

Disk utilization — Disk transfer statistics
: Display activities of block devices using the *sar -d* command

Formatting activities
: Display */var/adm/fmtlog* log file of formatting activities

8.6.2 File systems

This menu item gives you information on file system configuration, devices, file system parameters, disk utilization, mounted file systems and processes which are using a file system. You can also check file systems here:

FS configuration — Partitions
: Display partition arrangement from the */etc/vfstab* file

FS configuration — Devices — List
: List generated devices using the *getdev* command

FS configuration — Devices — By attributes
: List devices by attributes using the *getdev -a <attibutes>:** command

FS configuration — Devices — Device attributes
: Display attributes for a device using the *devattr -v <special_file>* command

FS configuration — Devices — Device mountpoint
: List mountpoints of devices using the *devattr <special_file> <mountpoint>* command

FS configuration — Devices — Table
: List table of device attributes from the */etc/device.tab* file

FS configuration — Devices — File system types1
: Display */etc/dfs/fstypes* log file for file systems

FS configuration — Devices — File system types2
: Determine file system type using the *fstyp <special_file>* command

FS configuration — FS parameters — Parameter1
: Output command line parameters using the *mkfs -m <device>* command

FS configuration — FS parameters — Parameter2
: Output command line parameters using the *newfs -N <device>* command

FS configuration — FS types
: Display generated file system types using the *crash << l; vfssw; l* command

FS configuration — Directories
: Display large directories using the *find . -type d -size +40 -print* command

Disk utilization — Disk space assignment
: Display number of free disk blocks and files using the *df -t* command

Disk utilization — I-Node assignment
: Display number of assigned and free i-nodes using the *df -i* command

Disk utilization — Utilization
: Display free disk space using the *dfspace* command

Disk utilization — Disk usage
: Display overview of disk usage using the *du* command

Disk utilization — File access
: Display use of system routines for file access usage using the *sar -a* command

Mount — Mounted file systems1
: Display mounted file systems using the *mount* command

Mount — Mounted file systems2
: Display table of mounted file systems from the *etc/mnttab* file

Mount — Partition
: Display partition in relation to mountpoint using the *devnm <mountpoint>* command

Processes
: Display processes which are using a file system using the *fuser <mountpoint>* command

FS check — Check1
: Check nonmounted file systems using the *fsck -m* command

FS check — Check2
: Check mounted file systems using the *fsck -l -n -F <fs_type> <block_device>* command

FS check — Reserved
: Display reserved devices using the *devreserv* command

8.7 System access

This menu item gives you information on the following:

- Entries in the *inittab* file
- Run level of the system
- Status of the *sac* process
- Network services control
- Port services for SAC
- Entries in the log files for SAC
- Line parameters for terminals
- Line settings for the TTY interface
- Description file for terminals
- Terminfo entries and the terminfo database
- STTY settings
- Contents of login files
- Contents of files for *su*
- System users

You can also check system user entries.

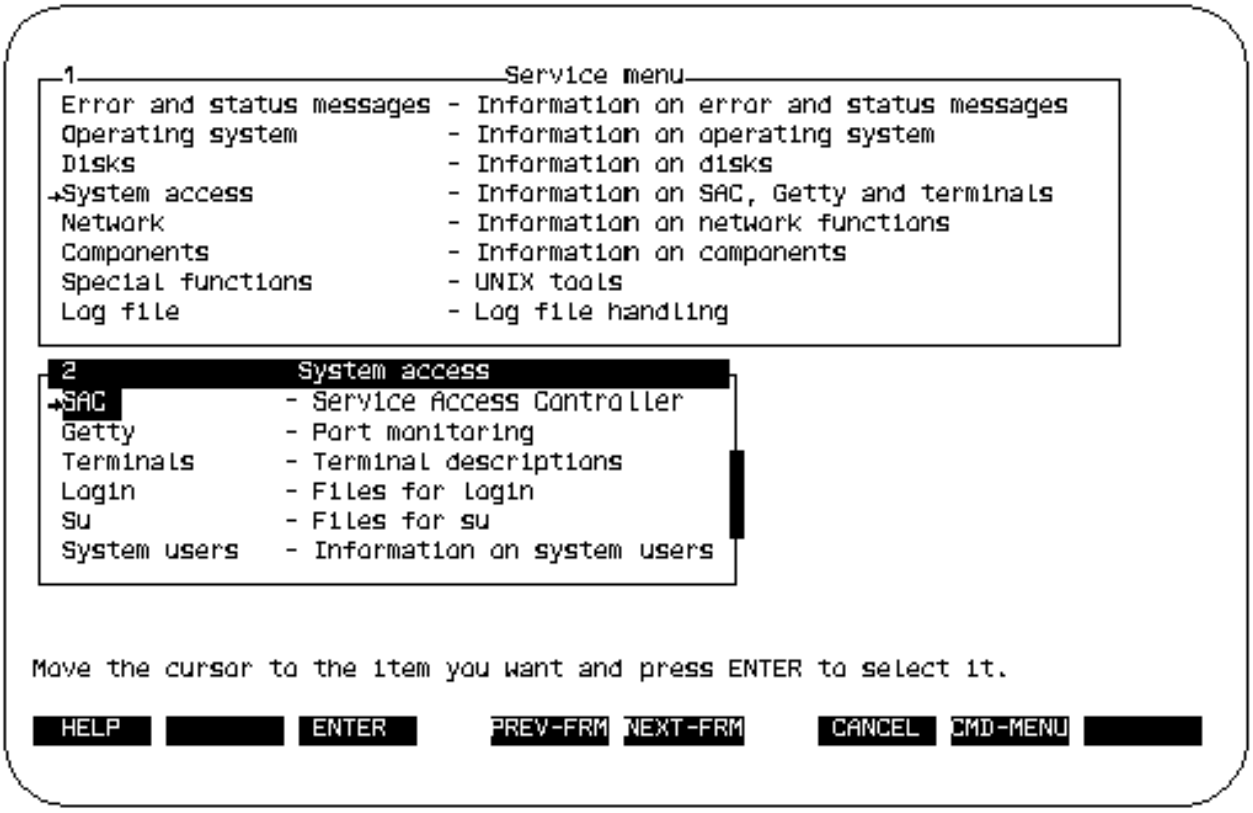


Figure 28: Diagnosis menu: System access

8.7.1 SAC

This menu item lgives you information on entries in the *inittab* file, the run level of the system and the status of the *sac* process (Service Access Controller). You can also look up information on network services control, port services for SAC and entries in the log files for SAC:

Inittab

Display entries in the */etc/inittab* file

Run level
Display run level of the system using the *who -r* command

SAC active
Display whether the *sac* process is running using the *ps -ef | grep sac* command

Network services controlling — Port monitor — Program
Display port monitors for a program using the *sacadm -f* command

Network services controlling — Port monitor — File
Display port monitors for a file from the */etc/saf/_sactab* file

Network services controlling — System-wide config script — Program
Display system-wide configuration script for a program using the *sacadm -G* command

Network services controlling — System-wide config script — File
Display system-wide configuration script for a file from the */etc/saf/_sysconfig* file

Network services controlling — P.m. config script — Program
Display configuration script for a program using the *sacadm -g -p <pmtype>* command

Network services controlling — P.m. config script — File
Display configuration script for a file from the *etc/saf/<pmtype>/_config* file

Port services — Program
Display port services for a program using the *pmadm -l* command

Port services — File
Display port services for a file from the */etc/saf/<pmtype>/_pmtab* file

Log files — SAC
Display */var/saf/_log* log file for SAC

Log files — Port monitor
Display */var/saf/<pmtype>/log* log file for port monitors

8.7.2 Getty

This menu item gives you information on the entries in the *inittab* file, the line parameters for terminals and the line settings for the TTY interface:

Inittab
Display entries in */etc/inittab* file

Gettydefs
Display line parameters for terminals from the */etc/gettydefs* file

Stty parameter — Program
Display STTY parameters for a program using the *sttydefs -l* command

Stty parameter — File
Display STTY parameters for a file from the */etc/ttydefs* file

8.7.3 Terminals

This menu item gives you linformation on the terminal description file, the terminfo entries, the terminfo database and the STTY settings:

Termcap
Display */etc/termcap* terminal capability description file

Terminfo — Description
Display terminfo descriptions for a terminal type using the *infocmp <terminal_name>* command

Terminfo — Data base
Display */usr/lib/terminfo* terminfo database

Stty — Allocation
Display allocation of terminal type and line from the */etc/ttytype* file

Stty — Stty parameters
Display STTY parameters for a line using the *stty -a <tty_name>* command

Stty — Search path
Display path for *<tty_name>* from the */etc/ttysrch* file

8.7.4 Login

This menu item gives you information on the contents of various login files.

Login logo
Display */etc/issue* file with the login logo

Default values
Display */etc/default/login* file for default login values

Failed login attempts
Display information on failed login attempts from the */var/adm/loginlog* file

Last login attempt
Display information on last login attempt from the */var/adm/lastlog* file

TERM for ttyname
Display allocation of *TERM* and *<ttyname>* from the */etc/ttytype* file

Global profile
Display global environment profile from the */etc/profile* file

System message
Display */etc/motd* message of the day file

Private profile
Display private environment profile from the *$HOME/.profile* file

Time zone
Display default system time zone from the */etc/TIMEZONE* file

8.7.5 Su

This menu item gives you information on the contents of the file for default *su* values and the log file for *su* calls:

Default values
: Display */etc/default/su* file for default *su* values

Log file
: Display */etc/var/sulog* log file for *su* calls

8.7.6 System users

This menu item gives you information on user IDs, group IDs and information on users who have logged in. You can also check system user entries:

Files — Users
: Display information on users from the */etc/passwd* file

Files — Groups
: Display information on groups from the */etc/group* file

Logged in users — Info1
: Display list of users and system information using the *logins -a* command

Logged in users — Info2
: Display extended list of users and system information using the *logins -x* command

Logged in users — Locally logged in
: Display users logged in to local host using the *who* command

Logged in users — Find user
: Display information on local and remote users logged in using the *finger* command

Logged in users — Remote logged in
: Display users logged in to remote hosts using the *rusers* command

Logged in users — Universally logged in
: Display users logged in to hosts on the local network using the *rwho* command

Logged in users — Password info
: Display user password information using the *passwd -s <user>* command

Logged in users — Last logged in
: Display last user to log in using the *last -<number>* command

Logged in users — Login info
: Display user login information using the *listusers* command

Test program — Users
: Check the *passwd* file using the *pwck* command

Test program — Group
: Check the *group* file using the *grpck* command

Test program — Display accounting records
: Display connection accounting records using the *fwtmp < /var/adm/wtmp* command

Test program — Check accounting records
: Check connection accounting records using the *wtmpfix/var/adm/wtmp* command

8.8 Network

This menu item gives you information on the following:

- TCP/IP status, configuration files, statistics and tests
- Routing tables and statistics
- NFS status, configuration files and statistics
- UUCP status, systems and configuration files

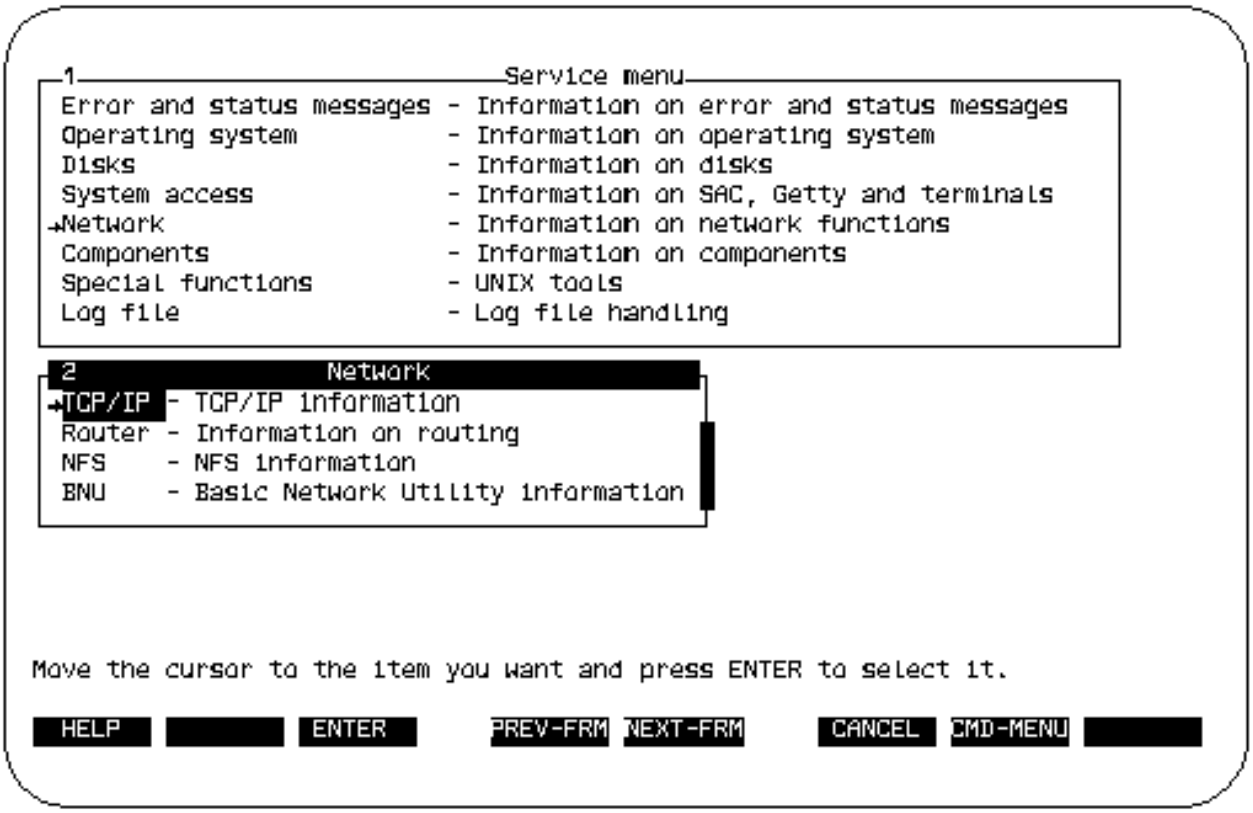


Figure 29: Diagnosis menu: Network

8.8.1 TCP/IP

This menu item gives you information on status details, configuration files, statistics and tests for TCP/IP:

Status — Network status
Display current network status using the *netstat* command

Status — Hardware interface
Display status of hardware interfaces using the *netstat -i* command

Status — Host status
Display host status of local computers using the *ruptime* command

Status — Name server inquiry
Look up name servers using the *nslookup* command


You start *nslookup* by entering *exit* and pressing the CR key when you see the prompt (>).

Status — Configuration file
Display configuration files from the */etc/inet* directory

Configuration — Address resolution
Resolve addresses between Internet and Ethernet using the *arp -a* command

Configuration — Inetd config
Display */etc/inetd.conf* configuration file

Configuration — Inetd services
Display */etc/services* services file

Configuration — Protocols
Display */etc/protocols* protocol file

Configuration — Network config
Display */etc/netconfig* network configuration file

Configuration — Header file
Display */usr/include/sys/netconfig.h* header file for network configuration

Configuration — Network names
Display */etc/networks* network names file

Configuration — System names
Display */etc/hosts* system names file

Configuration — Inetd config file
Display */etc/default/inet* configuration file

Statistics — Protocol statistics
Display protocol statistics for the network using the *netstat -s* command

Test — Connection test
Test connection to host using the *ping* command

Test — Send packages
Send packages to host and display them using the *ping -s* command

Test — TCP/IP connection
Monitor TCP/IP connection using the *trpt* command

Test — Login test
Test login to own system using the *rlogin localhost* command

8.8.2 Router

This menu item gives you information on the routing table and routing statistics:

Routing table
Display routing table using the *netstat -r* command

Routing statistics
Display routing statistics using the *netstat -rs* command

8.8.3 NFS

This menu item gives you information on the status, configuration files and statistics for NFS:

Status — Local resources
: Display available local resources using the *share -F nfs* command

Status — Remote resources
: Display available remote resources using the *dfshares -F nfs* command

Status — Mounted C resources
: Display mounted client resources using the *dfmounts -F nfs* command

Status — Mounted resources
: Display mounted file systems and remote resources using the *mount* command

Configuration — Commands
: Display commands for resource sharing from the */etc/dfs/dfstab* file

Configuration — File system types
: Display log file for supplied file systems from the */etc/dfs/fstypes* file

Configuration — Shared file systems
: Display table of shared file systems from the */etc/dfs/sharetab* file

Configuration — Host names
: Display */etc/hosts* file for host names

Configuration — Reliable Host S
: Display reliable hosts for the system from the */etc/hosts.equiv* file

Configuration — Reliable Host B
: Display reliable hosts for the user from the *~/.rhosts* file

Statistics — Network statistics
: Display network statistics using the *nfsstat* command

8.8.4 BNU

This menu item gives you information on the UUCP status, accessible UUCP systems, and the installation of various UUCP configuration files:

UUCP config check
: Check UUCP configuration using the *uucheck -v* command

UUCP status
: Display UUCP status using the *uustat -m* command

UUCP directory
: Display accessible UUCP systems using the *uuname* command

UUCP systems
: Display */etc/uucp* directory containing UUCP files

Terminal lines
: Display information on terminal lines from the */etc/uucp/Devices* file

System polling
: Display system polling from the */etc/uucp/Poll* file

BNU systems
: Display systems under BNU from the */etc/uucp/Systems* file

Dial-up codes
: Display shortcuts for dial-up codes from the */etc/uucp/Dialcodes* file

Access rights
: Display access rights for computer from the */etc/uucp/Permissons* file

Terminals
: Display list of terminals with dial-up passwords from the */etc/dialup* file

Dial-up passwords
: Display list of dial-up passwords from the */etc/d_passwd* file

8.9 Components

This menu item gives you information on the following:

- Installed software packages
- Software installation
- Printer spooler
- The *cron*-daemon

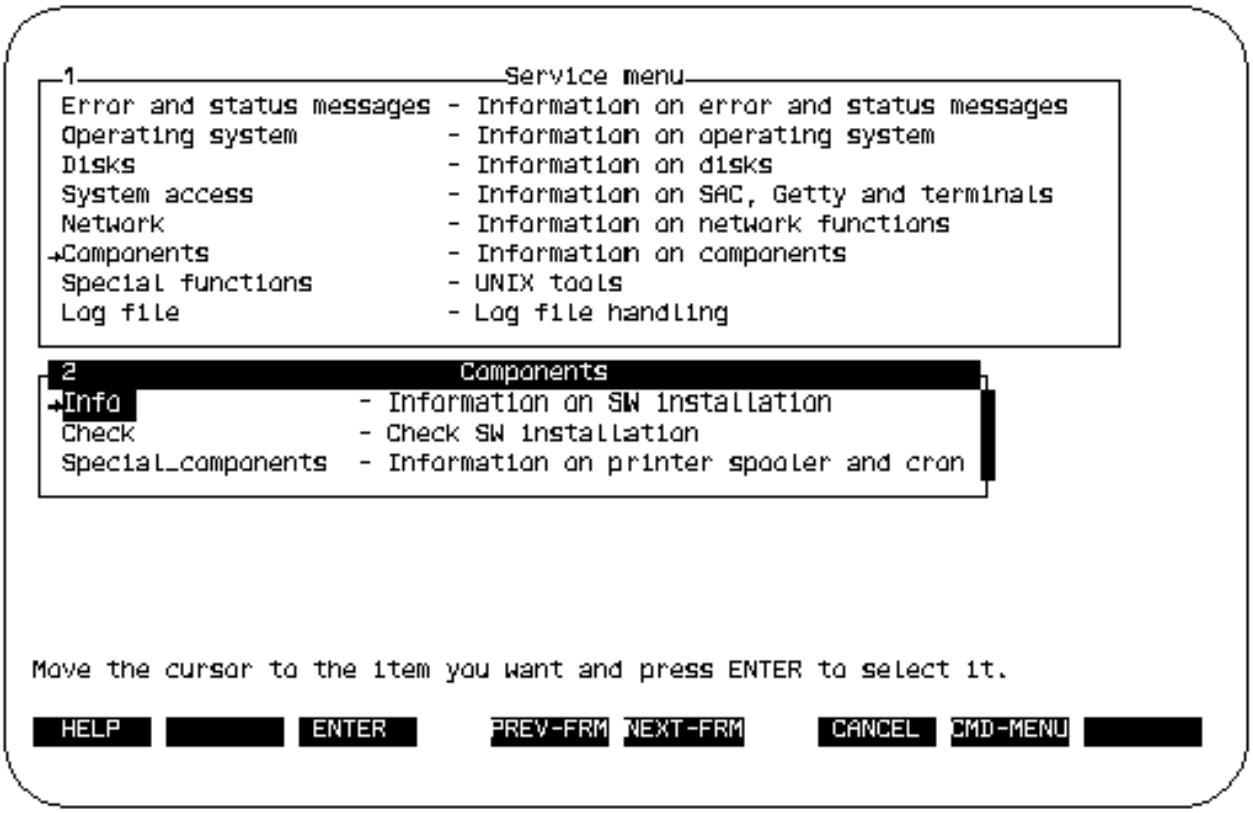


Figure 30: Diagnosis menu: Components

8.9.1 Info

This menu item gives you information on installed packages:

List Display installed packages using the *pkginfo* command

Information
Display detailed information on installed packages using the *pkginfo -l <package_name>* command

8.9.2 Check

This menu item gives you information on the list of files for an installation, and check the installation with and without editable files:

File list
Display list of files for an installation using the *pkgchk -l <package_name>* command

Installation check1
Check installation using the *pkgchk <package_name>* command

Installation check2
Check installation but not editable files using the *pkgchk -n <package_name>* command

8.9.3 Special components

This menu item gives you information on the printer spooler status and the *cron* daemon:

Printer spooler
: Display printer spooler status using the *lpstat -t* command

Cron — Config file
: Display */etc/default/cron* configuration file for *cron*

Cron — Cron active
: Check whether *cron* is running using the *ps -ef | grp cron* command

Cron — Cron files
: Display */etc/cron.d/FIFO* directory containing *cron* files

Cron — Cron tables
: Display tables for *cron* from the */var/spool/cron/crontabs* directory

Cron — At tables
: Display tables for *at* from the */var/spool/cron/atjobs* directory

Cron — Log file
: Display executed *cron* and *at* jobs from the */var/cron/log* file

Cron — Users
: Display *cron* table for user using the *crontab -l <user>* command

Cron — At job queue
: Display *at* job queue using the *atq* command

8.10 Special functions

You can do the following from this menu item:

- Display online manual pages using the *man* command
- Send mail to users using the *mail* command
- Change the password of the *diagnose* user using the *passwd* command

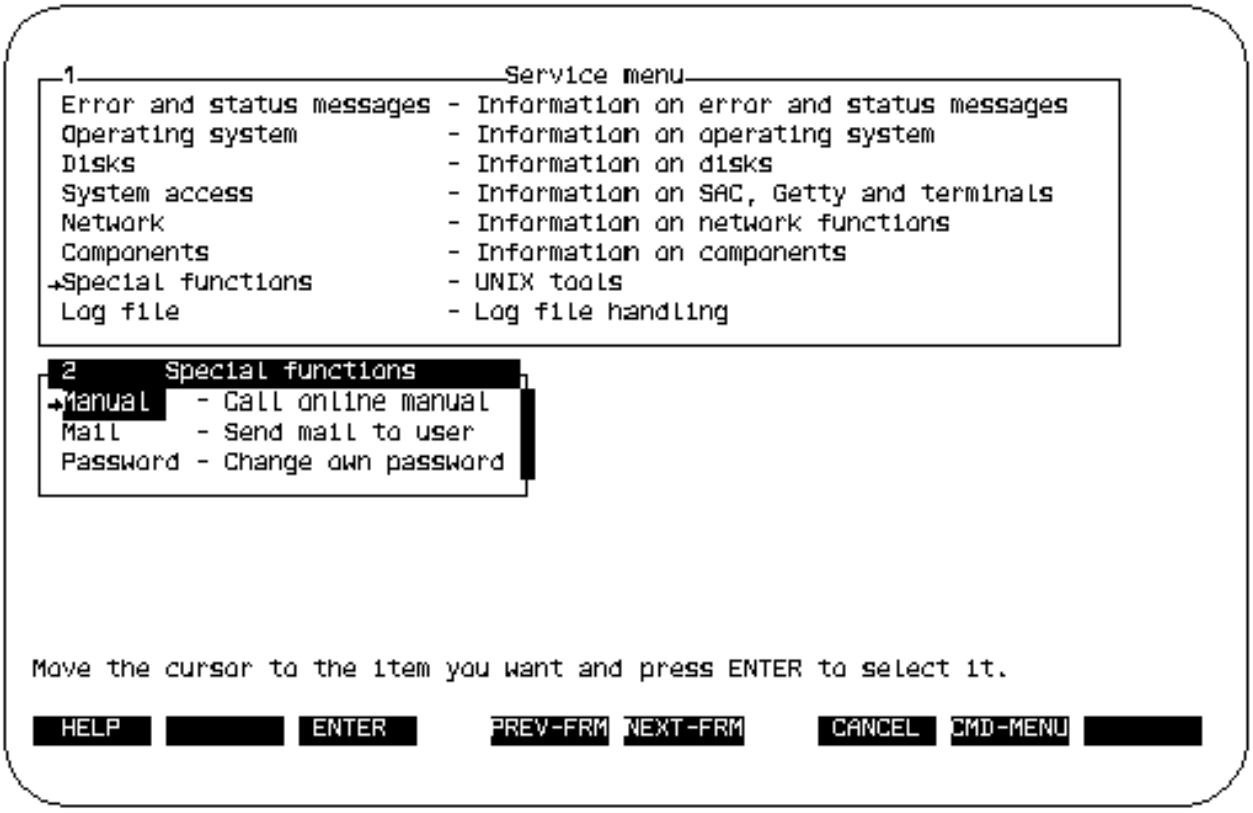


Figure 31: Diagnosis menu: Special functions

8.11 TDS2

The TDS2 package is associated with this menu item (see the chapter Test and diagnostic system TDS2 for monitoring the hardware).

8.12 Log file

You can do the following from this menu item:

- Display, print and clear the *proto2* log file in which command outputs were recorded
- Display print and clear the *proto* log file in which outputs were recorded as an option

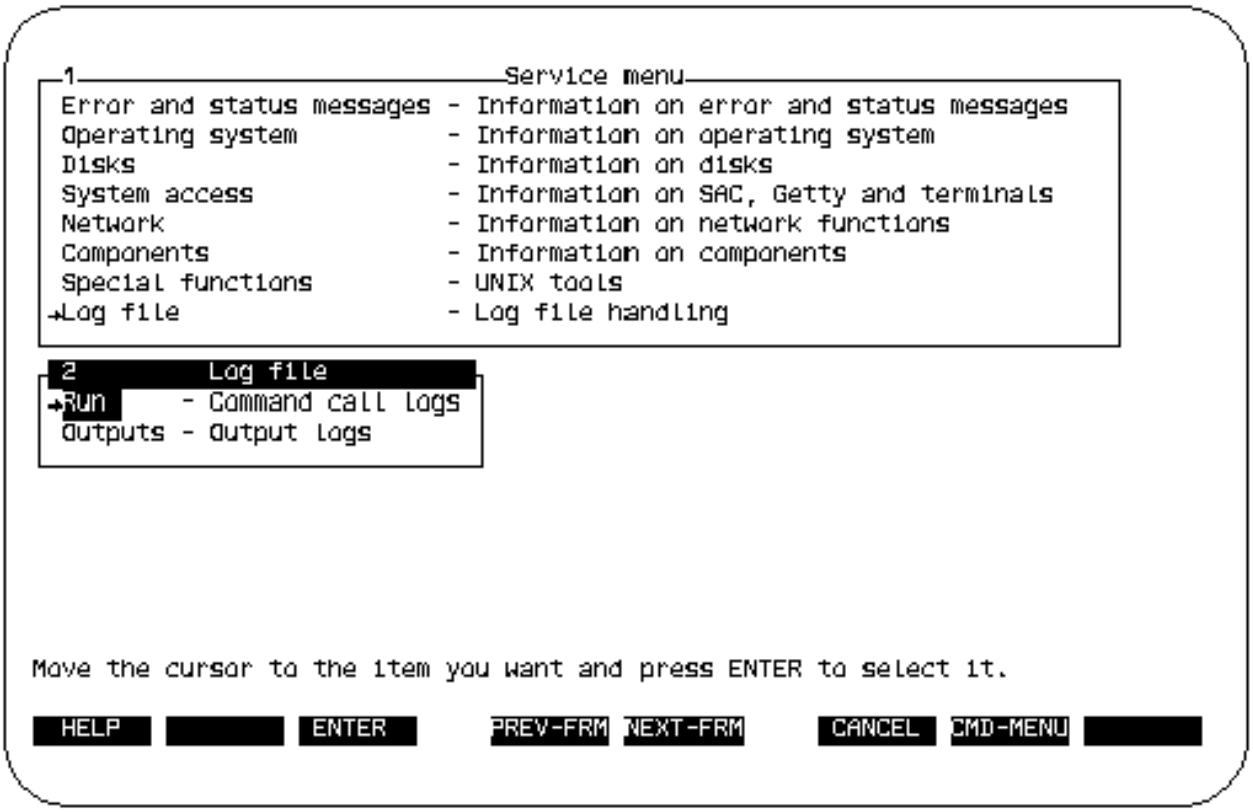


Figure 32: Diagnosis menu: Log file

The diagnosis menu has two log files in which details are recorded.

You have no influence on the creation of the first of these log files. It records all outputs and the command that generated them. The file is stored under */home/diagnose/proto2*.

Outputs from text windows can be recorded in the second log file as an option. If the PROTO softkey is pressed, the contents of the text window are written to the log file named */home/diagnose/proto*. The date, time and menu item that initiated the output is logged for each output.

8.12.1 Run

Display

Display *proto2* log file

Print

Print *proto2* log file

Clear

Clear *proto2* log file

8.12.2 Outputs

Display

Display *proto* log file

Print
: Print *proto* log file

Clear
: Clear *proto* log file

9 Test and diagnostic system TDS2 for monitoring the hardware

The test and diagnostic system TDS2 is a menu-driven testing utility for checking the system hardware, which runs under Reliant UNIX. The target group for TDS2 is the technical personnel of service companies or computer centers.

TDS2 is installed in the factory before the system unit is delivered, and will remain available even after an update installation. If the TDS2 is no longer present on your system you can reinstall it from the CD *CD-SYS-MI*.

You can start TDS2 in any of the following ways:

- Log on as the user *tds2*
- Call the menu item *tds2* from *sysadm*
- Call the menu item *tds2* from the diagnostics menu

You will find a description of this testing and diagnostic system in the Product Manual "Online Test System TDS2".

10 ServerView for the RM200/300/400

ServerView is the Server Management software supplied by Siemens AG. You can use *ServerView* to monitor all kinds of PRIMERGY servers and RM servers (RM200/ 300/400□C, RM300/400□E) within a network. The monitoring station is a standard PC running Windows□95/98 or Windows□NT. This software is supplied together with every PRIMERGY server. The agent for RM servers is included on the CD of Reliant UNIX System Tools (CD-SYS-MI).

With the help of its agents, *ServerView* monitors various functions on your servers, e.g. fans, processor load, temperature. Suitable corrective measures can then be initiated when specified values are exceeded or the system ceases to function correctly. Some of these measures can be activated automatically (e.g. adjusting the speed of the fans).

ServerView operates within heterogeneous networks using the network management protocol SNMP (simple network management protocol). This enables Siemens servers to be seamlessly integrated into company-wide network management systems.

This chapter will only give you a brief summary of *ServerView*. You will find detailed information regarding its installation, administration and operation in the *ServerView* User Guide.

10.1 The architecture of ServerView

ServerView comprises the monitoring station (the manager) and the agents.

The manager runs on a centralized monitoring system (standard Siemens PC) with a user interface running under Windows. All information relating to your servers' configuration and operational status is collated here, and from here you can govern their behavior in the event of a fault.

The agents are programs which run on the servers that are being monitored. They establish the information required by the manager and pass it on. The use of "traps" also enables the agents to inform the manager autonomously about particular events that occur on the servers. The agent in an RM system uses the EEC daemon for communications with the monitoring controllers.

Agents are available for Reliant UNIX, Windows NT, NetWare, OS/2 Warp and UnixWare.

10.2 The functions of ServerView

A modern server management system cannot do without an extensive, flexible and user-friendly functionality. *ServerView* has:

- comprehensive recovery functions for the rapid resolution of faults
- a powerful archive function for backing up the information on the management console
- early warning of threatened module failure
- monitoring functions for long-term monitoring of the servers
- extensive management functions, including excellent alarm management facilities with numerous option settings for the secure forwarding of messages
- the possibility of location-independent server management using the remote access service (RAS)

10.2.1 Monitoring functions

The monitoring functions of *ServerView* are used to inventory the installed hardware and permanently monitor the servers and the attached storage extension units. They enable you to obtain an up-to-date summary of the status of your servers at any time. The information is displayed in an intelligible form using clearly interpretable symbols.

The monitoring functions performed by *ServerView* exhibit the following features:

- Monitoring of all hardware components including metering the operating hours of the server (power-on times, power-off times and frequency of power-on)
- Clear display of servers and desktops in the network

  - redesigned, clear list of servers showing the features of their configuration
  - server summary with a status display in the summary bar
- Filtering of the list of servers: for instance, you can display only those servers for which one or more components have failed. This function increases the user-friendliness and intelligibility of ServerView, simplifying the recognition of faults and greatly facilitating the rapid pin-pointing of the cause of the fault.
- Display of detailed information about:
  - CPUs
  - Bus systems (EISA, PCI)
  - Hard disks
  - Main memory
  - Controllers
  - Operating system
  - Power supply
  - Voltages on the system board
  - Fan status
  - Ambient temperature and CPU temperature
  - Network cards
  - Board IDs
- Freely configurable polling cycles for individual optimization of query frequency and network load
- Storage of the received information for purposes of long-term monitoring (performance analyses, fault analyses

10.2.2 Management functions

ServerView offers well structured user-friendly centralized management tools, which you can use to adapt the system to your requirements. The system monitors specific parameters at prescribed intervals and keeps the administrator informed about the status of the servers in the network by means of a variety of alarm mechanisms.

Alarm Management

The alarm management provided by *ServerView* enables you to set up and manage a variety of alarms (traps) and forward them, for instance, to a pager (city call), an e-mail system or some other management system. All the set up functions are performed within just one uncomplicated window.

The straightforward and flexible configuration of alarm management and its wide variety of functions ensure that the administrator is quickly and reliably informed about any problems that arise.

The alarm management facilities of *ServerView* are particularly remarkable for the following functions:

- Grouping of alarms:
  - Convenient assignment of standard supplied alarm groups or user-defined ones to specifiable servers
  - Variable definition of alarm groups by free selection of traps
  - Linking of one or more alarm groups with a server group
  - Freely configurable forwarding of alarms for alarm groups
- Alarm manager:
  - Administration and analysis of the received alarms
  - Precise display of the time received, the source, the type of alarm and the forwarding mechanism used
  - Sortable by server, type of alarm and time
  - Possibility of dealing with all settings in only one window
- Alarm monitor:
  - Display of alarms
  - Recording of alarms
- Manifold mechanisms for forwarding alarms
  - SMS (short message service, i.e. messages can be forwarded to mobile phones and the networks D1 , D2 and E plus)
    - Secure transfers: messages are stored for up to 48 hours
    - Cost-effective (up to ten times cheaper than paging)
    - More secure from eavesdropping than pager messages
  - Forwarding of alarms via a mail system with MAPI interface
  - Numeric and alphanumeric paging (city call)
  - Launching of any desired program
  - Mailing (broadcast)
  - Forwarding of an alarm to some other management system, such as HP OpenView
  - Forwarding of alarms to other integrated systems, such as RemoteView, DeskView or APC PowerNet SNMP Manager for *ServerView*.
  - Writing of alarms into the NT Event Log. This enables you to display alarms without using the SNMP protocol.

Threshold Management

Threshold Management enables you to monitor limiting values, known as thresholds. This is particularly important when specific maximum or minimum values have to be adhered to during the operation of a server, for instance to prevent loss of performance or capacity bottlenecks.

Threshold Management offers:

- Monitoring of upper and lower limits as well as fluctuations
- Grouping of thresholds and convenient assignment of user-defined threshold groups to specifiable servers
- Freely configurable threshold values for monitoring parameters such as fan speed, etc.
- Displaying the specified threshold values for each server
- Link-up to alarm management via pre-set traps

Report Management

Report Management provides you with facilities for long-term monitoring of a variety of server parameters. This enables you, for instance, to recognize and document performance bottlenecks affecting the processor load or the network.

ServerView offers you the following functionality in this area:

- Long-term monitoring of servers (performance analyses, fault analyses)
- Trend analyses
- Configurable intervals
- Report list giving you an overview of all reports
- Data evaluated in graphical (linear or logarithmic) or tabular form.

11 Additional diagnostic resources

In addition to the diagnostic facilities provided by Logging □V3.0, VConfig or Config and the diagnosis menu, other resources are available to evaluate error messages.

11.1 Message files

If system initialization proceeded right through to the end despite some errors, then a basically functional system is available. Now you can evaluate the message files listed below to diagnose the problems that occurred:

/var/adm/log/osm

Kernel messages for the current session

The file */etc/.osm* which is used for kernel messages has been reorganized for Reliant UNIX V5.44. The default name and location for this file is now */var/adm/log/osm*.

This file will not be allowed to grow larger than about 5 MB. When this limit is reached, the existing file will be renamed to */var/adm/log/osm.<date>* and a new OSM file will be created. The maximum number of OSM files is normally set to 3.

If required, you can modify the file */etc/default/osm.conf* to change the name, maximum size and maximum number of OSM files.

/var/adm/log/messages

System logging messages

/var/adm/log/memory_messages

Main memory and cache error messages (only for RM 200 C, RM300 C and RM400 C)

/var/adm/error/errfile..Z*

Messages about recoverable SCSI device errors in the current or previous session (see also *zcat*(1) und *pyr_iosall*(8))

/var/saf/_log

Messages from the Service Access Facility

11.2 Trace flags for rc procedures

If the */etc/.trace.rc* and */etc/.trace.sd* files are empty and the operating state changes (including system startup and shutdown), then the names of the start and stop procedures are logged on the console before they are called.

The files are created as follows from a root shell:

- Start procedures (rc-scripts) with

  # > /etc/.trace.rc

- Stop procedures with

  # > /etc/.trace.sd

Subsequent messages or hangups can be localized better in this way.

11.3 SINIX/windows (RM200/300/400 only)

Under SINIX/windows you can evaluate hardware and software errors with a graphical user interface.

- You get to this interface in the Application Manager by means of *System — Diagnostics — Log Messages*.
- Under the Triteal Desktop (TED) you find the Application Manager in the drawer marked with a tool.

The messages are output as a list (3rd icon in the function bar) or as icons. Errors are classified by severity with traffic signs:

Alarm

Warning

Debug

Critical

Display

Miscellaneous

Double-clicking on the error message displays additional information in detail.

It is possible to arrange the error messages by various criteria (e. g. application, date, level etc.).

The built-in help function provides more information.

11.4 Logmsg (RM200/300/400 only)

Logmsg is used to log those system messages during the startup and shutdown phases which generally appear on the console only and are not subject to the normal logging mechanism.

Messages received by *logmsg* are stored temporarily in a file that is named */startup_log.tmp*, and can be passed to the Syslog daemon later with the */sbin/logmsg -copy* command.

Logmsg is set up by making the following modifications:

- In the */etc/inittab* file, replace all calls for */sbin/rc<n>* by */sbin/rc<n>_log* □ (*n*□=□0,1,2,3,6,S).

  Insert the following line between the *r6:*... and *co:*... lines:

  sl:23:wait:/sbin/logmsg -copy >/dev/console 2>&1

- Insert the */sbin/rc2* and*/sbin/rc3* scripts as the last line:

  echo ExIt LoGmSg

The next time the system starts up or the operating state changes to *2* or *3*, the messages will be stored first in the */startup_log.tmp* file together with the name of the calling procedure */sbin/rc<n>* (*n* = 0, 1, 2, 3, 6, S), a priority corresponding to the *LOG_NOTICE* level defined for *syslog*(3C), and a time stamp.

Afterwards, all messages are written to the */var/adm/log/messages* file by *syslog* with the facility *LOG_USER* and the level *LOG_NOTICE*, and can be evaluated with an editor or the diagnosis menu. The messages are also passed via the Syslog daemon to Logging.

12 Routines

Various routines are described in this chapter that may assist you in analyzing and resolving errors.

12.1 Switching to the board debug monitor (RM600)

The procedure you need to follow to switch to the board debug monitor depends on the current operational state of your system and the boot mode set.

- After the system has been turned on:
  - If autoboot is disabled (last digit of boot options is *8* or *9*), you can automatically go into the board debug monitor.
  - If autoboot is disabled (last digit of boot options is *1*), you can automatically go to the boot2 phase and from there into the board debug monitor (see below).
  - If autoboot is enabled, you can go to the boot2 phase by aborting the boot process with [CTRL][C] as soon as the message appears to load Reliant UNIX (e. g.):

    Autoboot: Waiting to load sdisk011s0/unix sd(1,8,0)unix□
    root=ios0/sdisk011s0 swap=ios0/sdisk011s1□
    (CTRL-C to abort, RETURN to expedite)

     The load device may differ depending on the configuration of your system.

    You can then go from the boot2 phase into the board debug monitor.
- From the boot2 phase or from Reliant UNIX:

  To go into the board debug monitor from the boot2 phase or from Reliant UNIX, use the following key combination:

  RM600 E
  : Press and hold down the [CTRL] key and type "@". Release the [CTRL] key, press the Shift key and enter the characters BDM (in upper case) directly afterwards (no blanks).

  RM600-xxx
  : Press [CTRL][A].

12.2 Switching to the PROM or firmware monitor (RM200/300/400)

The procedure you need to follow to go into the PROM or firmware monitor depends on the current operational state of your system and the boot mode set.

- After the system has been turned on:

  If autoboot is disabled, you can automatically go into the PROM or firmware-monitor.

  If autoboot is enabled, you can go into the PROM or firmware monitor by aborting the boot process with [CTRL][C] as soon as the message appears to load the SASH (e. g.):

  Autoboot: Waiting to load dkncr(0,0,10)sash□
  (CTRL-C to abort, RETURN to expedite)

   The load device may differ depending on the configuration of your system.
- From the SASH:

  From the SASH you can go to the PROM or firmware monitor by pressing [CTRL][D] or entering the SASH command *end*.
- From Reliant UNIX:

  From Reliant UNIX you can go to the PROM or firmware monitor with the *init□5* command (see *init*(1M)). If you are in multi-user mode, you should use the *who*(1) command before shutting down to check whether any other users are still logged in.

12.3 Loading the Mini System

In some error situations, the operating system cannot access the system disk. In such cases, you can draw upon a mini UNIX operating system which is on your Reliant UNIX 5.45 CD-ROM.

This section only gives a brief description of this. See the section "Loading the Mini System main menu" in the "Preparing for installation" chapter, in the "Reliant UNIX Installation and Operation" manual.

To load the Mini System from CD-ROM, you have to go into the board debug monitor on RM600 systems and to the PROM or firmware monitor on RM200/300/400 systems.

Proceed as follows to load the Mini System:

- Switch to the correct monitor, if you are not already there.
- Insert the CD-ROM in the drive.
- RM600:
  - Load boot2:

    BDM(MBII 0 t)>**b 1 cd(0,48,10)sash□□□□□(RM600-xxx)**
    BDM(SPB 8 t)>**b 1 cd(0,0,0)sash□□□□□□□□□(RM600 E)**

  - Load the Mini System:

    boot2: **install**

  Boot2 now searches for the first CD-ROM drive and asks if it is to be used for the load process:
  - Enter the address of the required CD-ROM drive or confirm the specified drive with the CR key.
- RM200/300/400:
  - On an alpha console, load the Mini System by entering the command:

    >> **install**

  - On a graphics console, the firmware menu now appears and you select the *Standard installation (install)* entry.

When the Mini System has booted and the CD-ROM drive has been mounted automatically, a dialog starts in which you have to define the language, console type and keyboard type.

The main menu of the Mini System then appears where you can:

- Perform a new installation

  You can perform a new installation of the operating system.
- Display information on an update installation

  You can display information on how to perform an update installation.
- Change to a shell in the Mini System

  You can temporarily exit the dialog program and get a shell in the Mini System, where you can then execute various Reliant UNIX commands, for example you can check the root file system or mount it to check the system files.

  You can return to the dialog program by entering the *exit* command or the key combination [CTRL][D].
- Read the readme file

  You can read the readme file which is contained on the CD-ROM.
- Write SASH to system disk

  You can rewrite the boot2 phase (RM600) or the SASH (RM200/300/400) to partition 10 of the system disk.
- Start DSSI

  You can start the *DSSI* data backup program.
- Abort Mini System and Reboot last system (RM600)
  Abort Mini System and Reboot (RM200/300/400)

  You can exit the mini system. The system is started from the system disk. The boot process ends in the board debug monitor (RM600) or in the PROM or firmware monitor (RM200/300/400).

12.4 Resolving inconsistencies in the NVRAM (RM600)

The values held in the NVRAM may become inconsistent for a variety of reasons (see the section Checking the NVRAM for inconsistencies).

If an inconsistency is detected in the NVRAM when the system is started up you will see a warning output on the console and the system start will be interrupted in the Boot2 sequence.

The NVRAM does not contain any correct data

In this case you will see the following warning:

```
NVRAM is not initialized or corrupt□
If you want to initialize the NVRAM use the command nv_init□
If you want to restore the NVRAM from a NVRAM-Backup on a systemdisk use the Command nv_restore□
boot2:
```

- Proceed as follows if a valid NVRAM backup exists on a system disk:
  - View the checksums of the NVRAM backups on the disks that are in the system cabinet (e. g.):

    ```
    boot2: nv_list□
    □
    List HIOS-NVRAM-Backups, please wait□
    HIOS-NVRAM:□□□□□□□NVRAM-CHECKSUM: 802a□
    ...□
    DISK: sd(1,0,0)□□□□□□□NVRAM-CHECKSUM: 908a□
    DISK: sd(2,0,0)□□□□□□□NVRAM-CHECKSUM: 908a□
    DISK: sd(1,8,0)□□□□□□□NVRAM-CHECKSUM: 908a□
    DISK: sd(2,8,0)□□□□□□□NVRAM-CHECKSUM: 908a□
    ...
    ```

  - If the checksums are identical on all disks, use any backup to restore the NVRAM, and continue system startup, e. g.:

    ```
    boot2: nv_restore sd(1,0,0)□
    boot2: boot
    ```

  - If the checksums are not identical, choose a disk which was in the system cabinet when the system was last shut down (the checksums of these disks are identical).

- If there is no valid NVRAM backup on a disk in the system cabinet, initialize the NVRAM and continue system startup:

  boot2: **nv_init**
  boot2: **boot**

The checksum of the NVRAM is not consistent with the checksums of the NVRAM backups on the system disk

In this case you will see the following warning:
WARNING: CHECKSUMS are different OLD_CHECKSUM: 1234 NEW_CHECKSUM: 1244
boot2:

- If the data in the NVRAM is inconsistent, please proceed as described under "The NVRAM does not contain any correct data".
- If there is no NVRAM backup, restore the NVRAM and continue system startup:

  boot2: nv_backup
  boot2: boot

The checksum of the NVRAM is not consistent with the checksum of the NVRAM backup on the boot disk

In this case you will see the following warning:
NVRAM_CHECKSUM and NVRAM_BACKUP_CHECKSUM of Rootdisk are different
boot2:

- If the warning is a result of one of the first two checks, proceed as described there.
- If the boot disk has been replaced, restore the NVRAM and continue system startup:

  boot2: **nv_backup**
  boot2: **boot**

12.5 Resolving inconsistencies in the NVRAM (RM200/300/400)

The values recorded in the NVRAM may be lost due to system errors, or by replacing the motherboard.

When Reliant UNIX is booted, if the system recognizes any inconsistency between the data held in NVRAM and the corresponding backup on the system disk, it will propose booting in single-user mode:

Checking channel_id configuration... Hit any key to go into single user mode

- Press a key and wait for the prompt.
- Use the *upd_channel_id*(8) command to correct the values in the NVRAM.
- Then switch to the PROM monitor

  init 5

- Boot the operating system:

  >> auto

When you boot the system, if the message cannot mount root is output, this may also be due to an inconsistency between the data recorded in the NVRAM and the corresponding backup on the system disk.

- Use the *checkchid* <*chidbootstring*> command to check whether there is an inconsistency. For <*chidbootstring*> you should enter the same value as for booting the SASH, but without specifying the file name *sash*, for instance:

  >> checkchid dkpcs(0,0,0,10)

- If the *checkchid* command does find an inconsistency, you can correct it using the *putchid* <*chidbootstring*> command. For <*chidbootstring*> you should enter the same value as for booting the SASH, but without specifying the file name *sash*, for instance:

  >> putchid dkpcs(0,0,0,10)

  The command *putchid* reads in a backup of the channel IDs from partition 10 of the system disk, which reconciles the NVRAM with the backup of the operating system settings.

- Then boot the operating system:

  >> auto

Replacing the motherboard destroys all the information that is stored in the NVRAM. Here's how to restore the NVRAM data:

- Before you replace the motherboard you must first establish the boot string! You do this by issuing the *printenv*(1) command:

  >> printenv bootfile

  When you have installed the new motherboard you will then have to tell the system this boot string again.

  It is always a good idea to keep a written record of the principal system data; once the motherboard has failed it will be too late to establish (for instance) the boot string!

- After you have replaced the motherboard, call the *putchid <chidbootstring>* command from the PROM monitor. For *<chidbootstring>* you should enter the same value as for booting the SASH, but without specifying the file name *sash*, for instance:

  >> putchid dkpcs(0,0,010)

  The *putchid* command reads in a backup of the channel IDs from partition 10 of the system disk.

- Use the *setenv* command to enter the old boot string, for instance:

  >> setenv bootfile dkpcs(0,0,010)sash

- Then boot the operating system:

  >> auto

You will find further information regarding assembling and removing components that are equipped for SIDATA in the Technical Manual of the relevant computer type.

12.6 Loading the old system kernel

There are a few cases where the current system kernel cannot be loaded properly. Such a case can occur, for example, if system parameters have changed in such a way that the system kernel has become too large or does not initialize properly. In such a case you have to boot the old system kernel in order to obtain an executable operating system. You can then investigate the reason for the defective kernel and resolve the problem.

The old system kernel can be loaded from the boot2 phase on RM600 systems and from the SASH on RM200/300/400 systems.

Proceed as follows to load the old system kernel:

- RM600:
  - If the system is in the boot2 phase and is awaiting input with the prompt *boot2:*, load the old system kernel as follows:

    boot2: **boot -f/unix.old**

  - If you are not in the boot2 phase, you have to load it from the board debug monitor.

    Go into the board debug monitor, if you are not already there, and load boot2:

    BDM(MBII 0 t)>**b 201 sdisk011s0/sash□□□□□(RM600-xxx)**
    BDM(SPB 8 t)>**b 201 sdisk000s0/sash□□□□□□(RM600 E)**

     The load device may differ depending on the configuration of your system.

    When loading in boot2 has been halted, load the old system kernel (see above).

- RM200/300/400:
  - If the system is in the SASH and is awaiting input with the prompt *sash:*, load the old system kernel as follows:

    sash: **boot -f dkncr(0,0,0)unix.old**

     The load device may differ depending on the configuration of your system.
  - If you are not in the SASH, you have to load it from the PROM or firmware monitor.

    Go into the PROM or firmware monitor, if you are not already there, and load the SASH:

    On an alpha console, load the SASH by entering the command:

    >> boot

    On a graphics console, the firmware menu now appears and you load the SASH by selecting the *Stand alone shell boot (boot)* entry.

    When loading in the SASH has been halted, load the old system kernel (see above).
- During the loading process, you are prompted to enter the name of the Reliant UNIX kernel again because the operating system expects the */unix* file. Enter */unix.old*.
- Log in as the *root* user and save the executable system kernel so that there will always be an executable kernel at the next power up:

  cp -p /unix.old /unix.sav
- Now take the necessary repair action. For example, deinstall packages or change the system parameters in the */etc/conf/cf.d/stune* file back to their original values with *idtune*(1M).
- Use *idbuild*(1M) to generate a new system kernel, and power up the system again with *init 6*.

12.7 System boot after system disk failure (RM600)

If your system disk is no longer available you can boot the system from a mirror disk or an alternative system disk, if you have one. Here's what you should do:

- Go into the board debug monitor and load boot2 with the physical boot string of the mirror disk or the other system disk, e. g.:

  BDM(SPB 8 t)>**b 1 sd(1,8,0)sash□□□□□(RM600 E)**
  BDM(MBII 0 t)>**b 1 sd(1,8,0)sash□□□□(RM600-xxx)**

  Or from the operating system CD-ROM:

  BDM(SPB 8 t)>**b 1 cd(0,0,10)sash□□□□□(RM600 E)**
  BDM(MBII 0 t)>**b 1 cd(0,48,10)sash□□□(RM600-xxx)**

- Take a look at the root partition table:

  boot2: **root -l**□
  □
  root_version: 2□
  entry□□state□□□□□log_part□□□□swap_part□□□phys_part□□disk_id□
  □□0□□□□active□□□□sdisk000s0□□sdisk000s1□□sd(0,0,0)□□SEAGATE□ST31200N□00217678□
  □□1□□□□inactive□□sdisk011s0□□sdisk011s1□□sd(1,8,0)□□SEAGATE□ST31200N□00166119□
  □□2□□□□undefined□
  state: undefined=0 inactive=1 active=2

- If there is no entry for the mirror disk or other system disk, generate it as shown in the following example:

  boot2: **root -n 1 -l sdisk011s0**□
  boot2: **root -n 1 -w sdisk011s1**□
  boot2: **root -n 1 -p sd(1,8,0)**

- Activate the mirror disk or other system disk in the root partition table:

  boot2: **root -n 1 -s 2**

- Go back to the board debug monitor and perform a cold start:

  BDM(SPB□8 t)>**creset□□□□(RM600 E)**
  BDM(MBII 0 t)>**creset□□□(RM600-xxx)**

   You can find more details on handling root mirror disks in the "Virtual Disks" manual.

If your system does not have mirror boot disks and no other system disk is available, you can only load the latest system backup or reinstall the system (See the "Reliant UNIX Installation" manual.).

12.8 System boot after system disk failure (RM200/300/400)

If your system disk is no longer available you can boot the system from a mirror disk or an alternative system disk, if you have one. You also have the option of restoring the system disk.

If the boot disk is no longer available then each time you boot the system will automatically stop in the PROM monitor or firmware monitor.

Booting from a mirror disk

If your system is equipped with a mirrored boot disk you can boot the operating system using this mirror disk.

 The following description applies to root mirrors that are maintained by the system (mroot). If you have configured a root mirror using a RAIDmaster controller you will be able to boot without taking any special action; the RAIDmaster controller will automatically use the functional half of the mirror.

- Use the firmware command *dkprint* to find out how the channel IDs correlate with the numbers assigned by

the firmware, for instance:

```
>> dkprint□
□
PCI-Path□□□□□□Channel-ID□□□□□□Bootstring□□□□□□□□□□□□□□□□□□□□Comments□
PCI#1/0□□□□□□□12□□□□□□□□□□□□□□dkpcs(0,0,0,10)sash□
PCI#1/1□□□□□□□13□□□□□□□□□□□□□□dkpcs(0,1,1,10)sash□
PCI#3.0□□□□□□□2□□□□□□□□□□□□□□dkdpt(0,0,0,10)sash□
PCI#3.1□□□□□□□3□□□□□□□□□□□□□□-□
PCI#4□□□□□□□□□-□□□□□□□□□□□□□□-
```

- If the column Channel-ID just contains a dash then no channel ID has yet been assigned. A dash in the column Bootstring signifies that this disk cannot be used as a boot disk.

- Boot the operating system. If the mirror disk in the above example is the disk with channel ID 13, then the Reliant UNIX designation of the mirror disk will be */ios0/sdisk131*. You can boot the system using the following command:

  >> boot -f dkpcs(0,1,1,0)/stand/unix root=ios0/sdisk131s0 swap=ios0/sdisk131s1

It is always a good idea to keep a written record of the principal system data. When you set up the root mirror you should therefore make a note of the logical name of the mirror disk.

If you are unable to set up a logical assignment you still have the possibility of moving the mirror disk to the slot occupied by the original disk; you can then boot the operating system without having to take any other action.

Booting from an alternative disk

If you do not have a mirror disk you can boot the system from a different system disk.

If you are unable to set up a logical assignment, execute the command *dkprint* as described in the previous section Booting from a mirror disk.

- Use the *boot <bootstring>* command to boot the SASH, e. g.:

  >>boot -f dkpcs(0,0,0,10)sash

  You can then boot the operating system, e. g.:

  sash: **dkpcs(0,0,0,0)unix**

You can now also use the command *setenv* to define the variable *bootfile* and launch the operating system directly from the PROM monitor. Since this procedure preserves the new value of *bootfile* it will then automatically be available for all subsequent system starts, e. g.:

>> setenv bootfile dkpcs(0,0,0,10)sash
>> auto

If the system has been modified in a way which has left it in an incompatible state (for example, if *upd_channel_id* has been used to exchange channel IDs), you will have to reconfigure the system disk. The following page explains how to do this.

Reconfiguring the system disk

If the previous suggestions have failed to solve the problem you still have the possibility of reconfiguring the system disk.

You will have to reconfigure the system disk if it is being given a different SCSI ID, or is being attached to a different SCSI string or a different controller.

- To reconfigure a system disk, first load the mini-system and from its main menu change to a shell.
- From this shell, call up the *rr* command (root recovery). This command will output a list in the following form:

  bootable system(s) found on:□
  /dev/ios0/sdisk054s0□□□dkdpt(1,0,4,0)□□□□□sash found□
  /dev/ios0/sdisk055s0□□□dkdpt(1,0,5,0)□□□□□unix but no sash

  sash found:
  : This signifies that the disk can be used as a boot disk, because *rr* has encountered a SASH and a UNIX kernel.

  unix but no sash:
  : This signifies that, before you can use this hard disk as a system disk you will first have to restore the SASH. See "Write SASH to system disk" in the section Loading the Mini System.

- Select the desired system disk (for example, */dev/ios0/sdisk054s0*).

  The appropriate nodes will then be set up, or modified, and the files and the NVRAM entries will be corrected. The system will then be rebooted. This procedure ensures that the system can get up as far as single-user mode.

The system administrator is then responsible for all the remaining modifications (e.g. */etc/vfstab*, */etc/dktab*).

If none of the measures described in this section are successful then you will either have to restore the most recent backup or install the operating system afresh.

12.9 Powering up in single-user mode

In some cases it may be necessary to power up the system in single-user mode in order to carry out certain administrative tasks. In single-user mode, only the root file system is mounted. Furthermore, the network is not available.

The system can be booted in single-user mode from the boot2 phase on RM600 systems and from the SASH on RM200/300/400 systems.

Proceed as follows to boot in single-user mode:

- RM600:
  - If the system is in the boot2 phase and is awaiting input with the prompt *boot2:*, boot the system as follows in single-user mode:

    boot2: **boot -f /unix initarg=-s**

  - If the system is not in the boot2 phase, you have to load it from the board debug monitor.

    Go into the board debug monitor, if you are not already there, and load boot2:

    BDM(MBII 0 t)>**b 201 sdisk011s0/sash□□□□□(RM600-xxx)**
    BDM(SPB 8 t)> **b 201 sdisk000/sash□□□□□□□□□(RM600 E)**

     The load device may differ depending on the configuration of your system.

    When loading in boot2 has been halted, boot the system in single-user mode:

    boot2: **boot -f /unix initarg=-s**

  - You can also go directly to single-user mode from Reliant UNIX by powering down the system as follows:

    cd /□
    shutdown -is -g0

- RM200/300/400:
  - If the system is in the SASH and is awaiting input with the prompt *sash:*, boot the system in single-user mode as follows:

    sash: **boot -f dkncr(0,0,0)unix initarg=-s**

     The load device may differ depending on the configuration of your system.
  - If you are not in the SASH, you have to load it from the PROM or firmware monitor.

    Go into the PROM or firmware monitor, if you are not already there, and load the SASH.

    On an alpha console, load the SASH by entering the >> *boot* command.

    On a graphics console, the firmware menu now appears and you load the SASH by selecting the *Stand alone shell boot (boot)* entry.

    When loading in the SASH has been halted, boot the system in single-user mode:

    sash: **boot -f dkncr(0,0,0)unix initarg=-s**
  - You can also go directly to single-user mode from Reliant UNIX by powering down the system as follows:

    cd /
    shutdown -is -g0
- As soon as you are in single-user mode, enter the root password:
  Type Ctrl-d to proceed with normal startup,
  (or give root password for system maintenance):

  After you enter the root password, the prompt for single-user mode is displayed:
  Entering System Maintenance Mode
- You can now carry out repairs on the system and then power up the system again with *init 6*.

13 I/O errors

In this chapter we will point out various errors that can occur in conjunction with the cabling and startup of peripherals such as printers, terminals and SCSI drives. The installation and startup of these components can be referenced in the following manuals:

- Operating Manual for the system unit
- Operating Manual for the peripherals
- System Administrator's Guide
- System Administration and Hardware Configuration Using the SYSADM User Interface (SCSI drives: RM200-xxx, RM400-xxx only)
- Hardware Configuration with Config under SINIX/windows (RM200-xxx, RM400-xxx only)
- User and Administrator's Guide for Xprint (for printers)

In relation to diagnostics, it is assumed that you have performed the installation in line with the instructions in these documents.

13.1 Printers

13.1.1 New connection problems

- The printer cannot be accessed.

  The printer is not ready.
  - Switch the printer online. The printer ready status is generally signalled by a LED on many printers.

  The power lead is not plugged in.
  - Check that all connectors are properly plugged in.

- The system and printer interfaces are not the same type physically.

  Although the plug-in connections sometimes look the same, they may be of different types. The following physical interfaces are available for printers on an RM system in its basic configuration or with the addition of optional controllers:
  - Parallel (Centronics / Bitronics)
  - Serial V.24
  - Serial V.11 (SS97)
  - IHSS (via 9766 terminals)
  - IMD (via converter to a printer with V.11 or V.24 interface)
  - AFP (via TAK to a printer with V.11 or V.24 interface)

  ▶ Refer to the respective operating manual to check if the interface types match.

  The power lead is not correct.

  ▶ Check whether you have used the correct type.

  There are any number of power leads that look identical at first glance but which have a different pin assignment internally. You also cannot assume that a lead that was used successfully with the printer on another computer type is suitable for the new connection.

  Every power lead that Siemens Nixdorf produces is identified by means of a reference number, e. g. T26139-Y2052-M5 or 51410.07.0.16. Refer to the operating manual for your system to check the correct cable types for the various attachments.

  The printer type is not suitable.

  ▶ Before you can connect up printers from third-party suppliers, you should have Siemens Nixdorf check whether the particular printer type is suitable. The following points need to be considered with regard to printers:
  - Has the printer an interface that allows it to be used with the RM system?
  - Is there a suitable cable type available for the printer?
  - Does Reliant UNIX support the printer protocol?
  - Is the printer supported in the *Xprint* spool system or is a replacement type available?

13.1.2 Problems during operation

- The printer produces "dirty" characters or reports parity errors.

  The data is corrupted at the interface.
  - The lead is too long. Please note the maximum lead length for the interface type you are using. Check the operating manual for your system unit for this information.
  - The lead is damaged or not properly shielded.
  - The lead is running parallel to power lines in the building. The data may be corrupted if surges occur.
  - The cable is not the right type. Some types of interface require twisted-conductor cables for data and signal cables.

  ▶ If you cannot locate the source of the error yourself, you should have one of our technical advisors check your cabling.

- Longer texts are printed garbled.

  The first section of the text does not contain errors. After that only fragments of text are correct. The size of the printer buffer determines how large the initial correct text section will be. You are operating the printer via a serial interface.

  The flow control is disrupted.
  - The power lead is not the right type (the ready signal or the printer send data does not reach the computer).
  - The flow control protocol is either set incorrectly in the printer setup or not set at all.

- Check the cable type in the operating manual or the possible settings in Setup.

- The printer signals it is ready to receive but the printer is not ready in the operating system.

  The signal cable for ready to receive is interrupted or not available.
  - Check that the cable is connected properly. If there is any damage, replace the cable.

- What is the status of the print job?

  For information about querying the status of printers and print jobs, refer to the User and Administrator's Guide for Xprint (Spool commands *xpstat* and *xpshow*).

13.2 Terminals

13.2.1 New connection problems

- No login prompt appears on the screen.

  The terminal is not configured correctly.
  - Check the configuration in Config. Inconsistent configurations are marked in the menus with an exclamation mark.

- The interface at the RM system and at the terminal are not the same type physically.

  Although the plug-in connections sometimes look the same, they may be of different types. The following physical interfaces are available for terminals on an RM system in its basic configuration or with the addition of optional controllers:
  - Serial V.24
  - Serial V.11 (SS97)
  - IHSS (9766 terminals)
  - IMD (via converter to a terminal with V.11 or V.24 interface)
  - AFP

  - Refer to the operating manual to check if the interface types match.

- The power lead is not correct.

  There are any number of power leads that look identical at first glance but which have a different pin assignment internally. You also cannot assume that a lead that was used successfully with the terminal on another computer type is suitable for the new connection.

  Every power lead that Siemens Nixdorf produces is identified by means of a reference number, e. g. T26139-Y2052-M5 or 51410.07.0.16. Refer to the operating manual for your system to check the correct cable types for the various attachments.
- The terminal type is not suitable.

  Before you can connect up terminals from third-party suppliers, you should have Siemens Nixdorf check whether the particular terminal type is suitable. The following points need to be considered:
  - Has the terminal an interface that allows it to be used with the RM system?
  - Is there a suitable cable type available for the terminal?
  - Does Reliant UNIX support the terminal type in the *terminfo*(4) database?

13.2.2 Problems during operation

- No login prompt appears after the system is turned on.

  The system was not booted in multi-user mode.
  - Check the system startup (this depends on the system unit, see chapters 2 or 3 as appropriate).

  The connection service for this connection is deactivated.
  - It can be activated under SYSADM by selecting the menu item *ports — port_services — enable*.

- No response on screen to key input.

  You may have entered the control character *XOFF* ([CTRL][S], [Pause]). This informs the computer that the terminal is not ready to receive.
  - Enter the key combination for *XON* control character ([CTRL][Q]).

  The power lead is interrupted.
  - Check the connector and the lead.

  The system process responsible for output on your terminal is no longer running or is not receiving CPU time because the system is heavily loaded.
  - If you can still access the system from another terminal or console, check if there are system bottlenecks.

- The function keys are not responding correctly.

  An unsuitable TERM variable has been defined for the connection. If you set up the connection in Config, the TERM variable will be set correctly. You can look at the current value using the *echo $TERM* command.
  - Check if one of the following causes is possible:
    - A different terminal was connected (adapt the configuration in Config).
    - A different emulation was selected in the terminal setup (select the original emulation).
    - The TERM variable will be modified on your system in a *.profile* or other shell procedure. (Try to pinpoint the position, e. g. check the files in question with the *grep TERM<Dateiname>* command and correct them if necessary).

  The [F1] ... [F5] function keys initiate local terminal functions.
  - Some terminal types, e. g. 97801-VT02, TC20, TC10 can initiate a local function for the first function keys. The application then doesn't notice the key being pressed. You can disable this local function in Setup. Please refer to the operating manual for the terminal for information on how to do this.

- There is no [Esc] key on the keyboard.

  On some keyboards, the ESCAPE function can only be accessed using a key combination.
  - On a VT keyboard, for example, press the [CTRL] and [3] keys at the same time.

- Cryptic characters appear on the screen.

  The transmission speed is set incorrectly.
  - Compare the terminal setup with the configuration in Config. The default setting for the console is 19200 bits/s and 38400 bits/s for the other V.24- or V.11 connections.

  A graphics character set is active. The normal character set is not enabled after an abnormal program termination. The *stty* parameters may also have been set incorrectly.
  - If only a graphics character set is active, enter the *ced dummy* command and exit the program immediately again with [CTRL][D]. The *ced* editor always loads an alphanumeric character set when it is exited.
  - If the *stty* parameters have been set incorrectly, enter the following command:

    stty sane

    (do not exit with the enter key but with [CTRL][J])

  The data is corrupted at the interface.
  - The lead is too long. Please note the maximum lead length for the interface type you are using (You will find this data in the operating manual.).
  - The lead is damaged or not properly shielded.
  - The lead is running parallel to power lines in the building. The data may be corrupted if surges occur.
  - The cable is not the right type. Some types of interface require twisted-conductor cables for data and signal cables.
  - If you cannot locate the source of the error yourself, you should have one of our technical advisors check your cabling.

13.3 SCSI drives

Problems during operation

- The peripheral cabinet is switched on but the integrated hard disks cannot be accessed in the system.

  The hard disks are not mounted.

  The hard disks are only recognized if they were mounted when the system was booted (high-availability configurations or RAID arrays allow hard disks to be mounted online using special procedures). To find out which hard disks are recognized in the system, use the *gettypes*(1) command.

  - Reboot the system with:

    shutdown -g10 -i6

  The SCSI connector is not plugged in.

  - Connect it and reboot the system.

- Errors such as SCSI-Parity Error, SCSI Power ON / Bus Reset Occurred are reported for the SCSI interface.

  The maximum permitted length for the SCSI bus is exceeded.

  - The length of the SCSI bus includes the internal and external leads. Furthermore, a length of 40 cm is added on for each plug-in connection in the SCSI bus. The total length must not exceed 6 meters.
  - Some configurations (e.g. peripheral cabinet connected to RAID CS31 Master Controller) allow the external lead to be a maximum of 150 cm long.

  - If you have extended the length of the SCSI bus using a longer cable or an additional peripheral box, please check if the error still occurs with shorter leads. You may need to have Service check the cabling and the devices.

The SCSI bus is not terminated properly.

- The SCSI bus must be terminated at both ends with a terminator. One can generally be found in the system unit and the other in the peripheral; in the case of the BG50 peripheral box, the terminator is located at the second SCSI connector. This error can occur if a terminator is missing or if there are more than two on the bus.

▶ If you have extended your system yourself, check the settings of the drives you have integrated.

A new drive has been switched on.

- After switching on a SCSI drive, a drive always returns with "SCSI Power ON / Bus Reset Occurred". These messages are suppressed, however, when the system is being booted. If you see these messages during operation, either a new drive has been switched on there is a loose contact in the power supply lead for the drive.

▶ Check the power cable in the peripherals box or the peripheral cabinet. If you cannot find an error, Service will have to resolve the problem.

14 Effects of a power failure

14.1 System configurations without BBU or UPS backup

In an unprotected environment, a power failure can have undesirable consequences for your data. The machine is immediately out of service and any data still in a cache is lost. All mounted file systems are then inconsistent.

In the subsequent boot process, the file systems are checked and repaired. In some cases, the repair cannot be carried out manually and you have to clean the file systems yourself in single-user mode using the *fsck*(1M) command. It may also happen, though rarely, that the file system cannot be repaired, in which case you have to restore the data from your backup.

You can avoid this inconvenience by using a BBU or UPS.

14.2 System configurations with BBU

The Battery-Backup-Unit bridges short-term power outages (approx. 3 sec) and allows the system to be brought down safely in the case of longer power failures. When a power failure is signalled by environment monitoring, system operation is terminated with a normal shutdown. You cannot continue working and the file systems are closed normally.

When power returns, the system automatically reboots. The BBU is then fully operational again once it is recharged; this normally takes approx. 36 hours if the battery is empty.

The service life of the batteries in the BBU is approximately 3-5 years, depending on the ambient temperature. In order to ensure that it functions properly, we would recommend that you have the battery replaced in good time.

14.3 System configurations with a UPS

Systems with the protection of a UPS allow operation to continue normally in the event of a power failure, You should remember, however, that peripherals, terminals, modems etc., that you require for operation, must also be connected via the UPS. The UPS daemon in the RM system routinely checks the status of the UPS and initiates a shutdown if the power failure lasts longer than a defined period of time. Once the power returns, the system is either booted immediately or after a defined charging time. On computers of type RM600 you can specify these times and govern the behavior of the system on shutdown or booting by modifying the *_upstab*(4) file, and on computers of type RM200/300/400 you can modify these settings using the *ups*(1M) command.

The duration of UPS operation depends on the capacity of the UPS and the gross output of the connected user. Please refer to the respective operating manual for information here.

In exactly the same way as with a BBU, the batteries in the UPS must be exchanged after a number of years.

Information on installing the UPS and checking the functions can be found in the "Reliant UNIX Operation" manual.

14.4 Redundant power packs

Some devices contain redundant power packs. If a power pack fails in this type of system, the other power pack takes over operation without any loss of functionality. The fault reporting mechanism provides information on the failure of the power pack.

If you connect the two power packs at different phases, the reliability of your system operation will be improved should a circuit fail.

When the primary supply returns or if you replace the power pack, a message is likewise issued.

Some devices allow the power pack to be replaced online, see the respective operating manual for further information in this regard.

15 The LAR concept

15.1 LAR on computers of type RM600

It is important to be sure that, even if the individual hardware components of a system fail or malfunction, no single component is ever in a position to paralyze the operation of the entire system.

When the system is booted, it recognizes any defective boards or board components straightaway, during the BIST, and refrains from taking them into operation. In this way the system can still function in spite of a defect. The boot is aborted only if certain specific boards or board components are defective such that the system cannot operate at all (see the section Activating the boards).

However, this function only covers faults in boards that are detected in the BIST, and not disorders which may arise in boards, components of boards (e.g. controllers) , SCSI devices, BBUs, power units or fans in the course of normal operation, nor can it deal with overheating. The LAR concept (LAR = Lockout Auto Recovery) has been developed to deal with this problem, to enable these devices and the ambient temperature to be monitored while the system is in operation.

The monitoring of magnetic tape drives refers only to the cleaning interval and not to malfunctions involving an error or fault.

For information regarding *VConfig*, *Config* and *HW_CONFIG*, which are mentioned a number of times in this chapter, please refer to the chapter Hardware diagnostics using VConfig and Config and the section Outputting the HW_CONFIG defects table.

15.1.1 LAR for boards, components of boards and SCSI devices

15.1.1.1 Flagging faulty and defective hardware

Faulty and defective boards, components of boards and SCSI devices are flagged automatically when a fault or defect is recognized in response to one of the events that are specified for event management by the logging function (see the section Event Management).

You can also set such flags yourself, by hand, for instance if you have reason to suspect that a device is faulty:

- You can use the VConfig function *Deactivate* to give boards, components of boards and SCSI devices the flag defect.
- You can also use the shell command *set_lar*(1M) to give boards and components of boards the flag *error* or *defect*.
- You can also use the shell command *deact_dev*(1M) to give SCSI devices the flag *error* or *defect*.

The operating system makes no distinction between flags which are set automatically and those you set yourself.

Defective boards and components of boards are also recorded in NVRAM with the flag *defect*.

It may happen that the BIST fails to recognize sporadic faults which affect boards during normal operation. Since this kind of fault can adversely affect the operation of the computer, it will be advisable to prevent such boards from being taken into operation at the next boot. You do this by recording the *defect* flag in NVRAM for the relevant boards. Note that the *defect* flag will also be recorded in NVRAM for boards which do not pass the BIST (status FAIL). If the fault that was recognized during the BIST occurs intermittently, the board might otherwise successfully complete the BIST on the next boot and be taken into operation.

Faulty or defective boards, components of boards and SCSI devices are also recorded in the HW_CONFIG database with the attribute *error* or *defect* and displayed by VConfig with status *error* or *defect*.

15.1.1.2 Clearing the flags on faulty and defective hardware

You can clear *error* and *defect* flags manually in the following ways:

- You can clear flags from boards, components of boards and SCSI devices under VConfig using the function *Activate*.
- You can also clear flags from boards and components of boards using the shell command *reset_lar*(1M).
- You can also clear flags from SCSI devices using the shell command *act_dev*(1M).

There is no point in clearing a flag by hand unless you are sure than the fault has been cleared, or if the flag was originally set by hand.

The flags *error* and *defect* will be cleared automatically under the following circumstances:

- Flags on boards and SCSI devices which carry serial numbers will be cleared automatically when the system is booted if the component was replaced prior to the boot. The system recognizes this when the serial number changes. You can use the commands *showconf*(8) (only for boards) and *autoconf*(8) to see which of the boards and SCSI devices in your system carry serial numbers.
- Flags on SCSI devices which can be replaced using the OLR function are cleared when the replacement has been effected.

  Providing certain hardware and software requirements are met, you can use the OLR function (OLR = on-line replacement) to replace SCSI devices while the system is operation. Please refer to the operating instructions for your computer for details of these requirements.
- The cleaning flag on a magnetic tape device is cleared when the device has been cleaned.

15.1.1.3 Deactivating defective hardware

Boards or board components which are flagged as *defect* will be deactivated the next time the system is booted.

SCSI devices which are flagged as *defect* will be deactivated as soon as the flag is set, unless they are currently in use (open()). SCSI devices which are in use will be deactivated the next time the system is booted.

When they have been deactivated, these boards, board components and SCSI devices will also be flagged in the HW_CONFIG database as *deactive* and shown with status *deactive* in VConfig.

15.1.1.4 Activating deactivated hardware

Deactivated boards, components of boards and SCSI devices are reactivated as follows:

- Boards and SCSI devices which carry a serial number and were replaced after the system had been shut down, will be reactivated automatically the next time the system is booted. The system recognizes this by the serial number having changed. You can use the commands *showconf*(8) (only for boards) and *autoconf*(8) to see which of the boards and SCSI devices in your system carry serial numbers..
- Components of boards, and boards and SCSI devices which do not carry a serial number, have to be reactivated by hand using the VConfig function *Activate*.

  You should only reactivate components by hand after the defective hardware has been replaced, or when you are certain that the fault has been cleared, or when it was deactivated in response to a *defect* flag which was set manually.
- SCSI devices that are replaced using the OLR function are activated automatically when the replacement has been completed.

The activation of boards and components of boards never takes effect until the system is rebooted. On SCSI devices it takes effect straightaway.

15.1.2 LAR for BBUs, power units and fans

The system environment monitoring performed by the Ubft software automatically sets flags on BBUs, power units and fans and deals with any subsequent reaction.

Defective BBUs and power units are recorded in the HW_CONFIG database with the attribute *defect*, displayed on a red field in VConfig and displayed by Config with status *Defect*.

If a redundant fan becomes defective it is entered in the HW_CONFIG database with the attribute *defect*, displayed on a yellow field in VConfig and displayed by Config with status *Single_failure*.

If all the fans are defective they are entered in the HW_CONFIG database with the attribute *deactive*, displayed on a red field in VConfig and displayed by Config with status *Multi_failure*.

In each case the flags will be cleared automatically as soon as a defect has been corrected (or, at the latest, the next time the system is rebooted).

15.1.3 LAR for temperature problems

The system environment monitoring performed by the Ubft software automatically sets flags when there are problems with the temperature, and deals with any subsequent reactions.

Problems with the temperature are recorded in the HW_CONFIG database with the attribute *error* (non-critical) or *defect* (critical). Critical problems with the temperature are displayed in Config with status *Temperature_failure*.

The flags will be cleared automatically as soon as the problem has been dealt with (or, at the latest, the next time the system is rebooted).

15.2 LAR on computers of type RM200/RM300/RM400

It is important to make sure that, even if the individual hardware components of a system fail or malfunction, no single component is ever in a position to paralyze the operation of the entire system.

The LAR concept (LAR = Lockout Auto Recovery) has been developed to deal with this problem. It monitors the following components:

- CPU (only on the RM400 E)
- Main memory (only on the RM300□E, RM400□C and RM400□E)
- SCSI devices
- BBUs, power units and fans
- Failure of the primary power supply and overheating

When the system is booted, it recognizes any defective main memory modules (RM300□E and RM400□E) and CPUs (RM400□E) and refrains from taking them into operation. In this way the system can still function in spite of the defect.

You will find more information regarding *VConfig*, *Config* and *HW_CONFIG*, which are mentioned a number of times in this chapter, in the chapter Hardware diagnostics using VConfig and Config and the section Outputting the HW_CONFIG defects table.

15.2.1 LAR for CPUs and main memory modules

15.2.1.1 RM400□E and RM300□E

If the self-test or the operating system detects a defect in a CPU (RM400□E only) or a main memory module the faulty component will be recorded in a list of defects. Components that are on this list will not be taken into operation when the system is subsequently booted.

A message output during the boot procedure reports any changes that have occurred since the previous run of the system, e.g.:

modul: WARNING: *error number* The lar_defects for CPU's have been changed since last boot□
Old: 0x0000000000000100 New: 0x0000000003800000

The procedure */etc/rc2.d/S70lar* then records the defective components in the defects table of the HW_CONFIG database with the attribute *defect* and displays them in VConfig with status *defect*.

The defective components will automatically be deactivated during the next reboot. Deactivated components are not reactivated automatically at reboot after the fault has been cleared (e.g. by replacement).

If faults are reported in your system CPU or main memory you should contact the Service department. When the repair has been carried out the technician will reset the entry in the defects table.

You can modify the defects table yourself using the command */usr/bin/lar/updmemcpu*(8).

15.2.1.2 RM400□C

Faulty or defective main memory modules are flagged automatically when a fault or defect is recognized in response to one of the events that are specified for event management by the logging function (see the section Administering event definitions).

You can also use the command *deact_ctlr*(1M) to set these flags yourself (e.g. because you suspect a fault or defect) and the command *act_ctlr*(1M) to reset them.

Faulty or defective main memory modules are recorded in the HW_CONFIG database with the flag *error* or *defect* and displayed in VConfig with status *error* or *defect* respectively.

Defective main memory modules are not automatically deactivated during the next reboot. Deactivated main memory modules are not reactivated automatically at reboot after the fault has been cleared (e.g. by replacement).

15.2.2 LAR for SCSI devices

15.2.2.1 Flagging faulty and defective devices

Faulty and defective devices are flagged automatically when a fault or defect is recognized in response to one of the events that are specified for event management by the logging function (see the section Event Management).

You can also use the command *deact_dev*(1M) to set such flags yourself, for instance when you suspect that a device is faulty. The operating system makes no distinction between flags which are set automatically and those you set yourself.

Faulty or defective devices are recorded in the HW_CONFIG database with the attribute *error* or *defect* and displayed by VConfig with the status *error* or *defect*.

15.2.2.2 Clearing the flags on faulty and defective devices

You can clear *error* and *defect* flags manually using the shell command *act_dev*(1M).

There is no point in clearing a flag by hand unless you are sure than the fault has been cleared, or if the flag was originally set by hand.

Flags on devices that are replaced using the OLR function are cleared automatically when the replacement has been effected.

Providing certain hardware and software requirements are met, you can use the OLR function (OLR = on-line replacement) to replace SCSI devices while the system is operation. Please refer to the operating instructions for your computer for details of these requirements.

15.2.2.3 Deactivating defective devices

Devices that are flagged *defect* will be deactivated as soon as the flag is set, unless they are currently in use (*open()*). Devices that are in use will be deactivated the next time the system is booted.

When they have been deactivated, these devices will also be flagged in the HW_CONFIG database as *deactive* and shown with status *deactive* in VConfig.

15.2.2.4 Activating deactivated devices

Deactivated devices are reactivated as follows:

- Devices which have serial numbers and are replaced after powering down the system will be automatically activated when the system is rebooted. You can use the command *autoconf*(8) to see which of the devices in your system carry serial numbers.
- Devices without serial numbers have to be activated by hand using the command *act_dev*(8).
- Devices that are replaced using the OLR function are activated automatically when the replacement has been effected.

The devices can be accessed again as soon as they have been activated.

15.2.3 LAR for BBUs, power units and fans

The system environment monitoring performed by the EEC daemon automatically sets flags on BBUs, power units and fans and deals with any subsequent reaction.

Defective BBUs and power units are recorded in the HW_CONFIG database with the attribute *defect*, displayed on a red field in VConfig and displayed by Config with status *Defect*.

If a redundant fan becomes defective it is entered in the HW_CONFIG database with the attribute *defect*, displayed on a yellow field in VConfig and displayed by Config with status *Single_failure*.

If all the fans are defective they are entered in the HW_CONFIG database with the attribute *deactive*, displayed on a red field in VConfig and displayed by Config with status *Multi_failure*.

In each case the flags will be cleared automatically as soon as the defects have been corrected (or, at the latest, the next time the system is rebooted).

15.2.4 LAR for temperature problems and failure of the primary power supply

The system environment monitoring performed by the Eip and Ptf software automatically sets flags when there are problems with the temperature or failure of the primary power supply, and deals with any subsequent reactions.

Problems with the temperature are recorded in the HW_CONFIG database with the attribute *error* (non-critical) or *defect* (critical). Critical problems with overheating are displayed in Config with status *Temperature_failure*.

Provided the relevant cabinet is equipped with a BBU or UPS, failure of the primary power supply is recorded in the HW_CONFIG database with the attribute *defect* and displayed in Config with status *Defect*.

The flags will be cleared automatically as soon as the problem with the temperature or the power supply has been dealt with (or, at the latest, the next time the system is rebooted).

16 Boot1 error messages on RM600 systems

This chapter lists all errors that can occur in the boot1 phase on RM600 systems (see also the section The boot1 phase). This list is sorted in ascending alphabetical order and makes no distinction between minor and serious errors because termination of the system startup process through a serious error is always the result of one or more minor errors beforehand and can only be interpreted in conjunction with them. Nor are the errors grouped by system (RM600-xxx or RM600□E) because most of them can occur on both systems. The following aspects indicate system-specific messages.

- Messages referring to the CSI controller apply to RM600-xxx systems.
- Messages referring to the HIOS or EHIOS controller apply to RM600□E systems.

The explanations of the individual errors are mainly aimed at Service engineers.

Boot2 got loaded

Boot mode *1000* has been set.

Bootstring is empty

The boot string does not contain any data.

Can't find my CPU/CPM/CP4 agent id (<*dec_value*>) in config list

The required CPU was not found in the configuration table.

Cannot enable function LAN

The CSI controller could not be initialized.

Cannot enable function SCSI

The CSI controller could not be initialized.

Cannot enable function TTY

The CSI controller could not be initialized.

Cannot enable function WARTUNG

The CSI controller could not be initialized.

Cannot find appropriate cabinet

Multibus configuration data was not found.

Cannot find configuration description

The configuration table has not been loaded into memory.

Cannot find CSI/DCS controller

The configuration table in memory is incorrect.

Cannot find file *'<file_name>'* on CD-ROM

The file name in the boot string is incorrect or the wrong CD-ROM has been inserted.

Cannot find file *'<file_name>'* on disk

The file name in the boot string is incorrect.

Cannot find MBP description

The *c_cons* entry is missing in the configuration structure. The CSI controller has not sent any configuration data, or the data that was sent is incorrect.

Cannot identify device for file *'<file_name>'*

The loading device specified in the boot string is not known.

Cannot initialize CSI

The CSI partition and CSI controller could not be initialized.

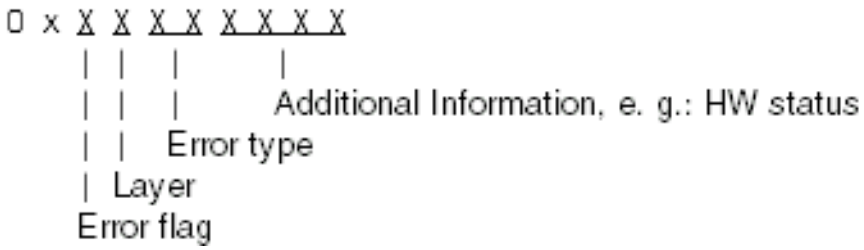
Cannot load LAN file (status: <*dec_value*>)

The required file cannot be loaded via Ethernet.

Status:	Status:
0 Status ok	14 Bootp bind failed
1 File is not in executable format	15 No server for this file
2 File has bad magic number	(server dont know the client
3 File has bad ELF identification	ethetnet
4 Short read – cannot load program	address, or bad host name, or no
headers	16 such file on server)
5 Short read – cannot load section	17 Unable to set interface address
6 Short read – cannot load section	18 File not found on server
7 Controller number is too large	19 Bootp open: tftp connect failed
8 File name is too long	20 Assertion failed
9 Unmatched parenthesis in file name	21 Sizeof(mbuf) not long aligned
10 Host name is too long	22 Not enough memory for mbufs
11 File name is too long	23 Free socket screw up
12 Internet address is not valid	24 IP packet is too large to xmit
13 Net address is incorrect	99 Timeout waiting for command end
	Any other error

Cannot load LAN file from X86 Ethernet controller (status: <*hex_value*>)

The desired file cannot be loaded over Ethernet. The hex number is made up as follows:



Layer:	Layer:
1 Controller	5 ARP (Address Resolution Protocol)
2 DLC (Data Line Control)	6 BOOTP (Bootstrap Protocol)
3 IP (Internet protocol)	7 TFTP (Trivial File Transfer Protocol)
4 UDP (User Datagram Protocol)	8 Hardware Init

Error type:	Error type:
01 Timeout	20 Invalid port
02 Read timeout	21 Invalid size
03 Max retry reached	22 Invalid Id
04 Controller busy	23 Invalid ethertype
05 Lockbyte set	24 Invalid checksum
06 SPbus init failed	25 Invalid protocol
07 Buffer shortage	26 Invalid sequence
11 Unsupported hardware	30 Missed data block
12 Unsupported function	31 File not found
13 Unsupported header	32 Access violation
14 Unsupported fragment	3F Error frame received

Cannot open file *'<file_name>'*

The file on the loading device cannot be opened. The drive is defective or not ready, or incorrect parameters have been specified in the boot string.

Cannot read boot2

The remaining blocks of Boot2 could not be read. The drive or the storage medium is defective.

Cannot read CD-ROM block

The CD-ROM volume header could not be read. The drive is defective, or no CD-ROM has been inserted, or the wrong CD-ROM has been inserted.

Cannot read disk block 0

The disk volume header could not be read. The drive is defective.

Cannot read ELF file

The first block of the program to be loaded could not be read. The drive is defective or not ready, or the wrong removable medium has been chosen, or the disk is not a system disk.

Channel on DCS board not connected

The required channel on the DCS board has not been assigned.

CSI already initialized

A system warm start has been performed.

CSI error <*hex_value*>, error code <*hex_value*>

The SCSI controller on the CSI board could not be initialized.

Device *'<device_name>'*: init failed - deconfigured

The drive could not be initialized.

Device is already open

A system warm start has been performed.

Device is always busy

Drive initialization has not been completed.

Double connection to SCSI bus <*dec_value*>.□
Check the SCSI cables□□□□# SCSI bus 0 (SE)□□□□□□□□□□□# SCSI bus 0 (SE)□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□# SCSI bus 1 (DE)□□□□or□□□□# Chain doubling□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□# Terminator bus 2□□□□□□□□□□# Chain doubling□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□# SCSI bus 2 (DE)□□□□□□□□□□□# SCSI bus 1 (DE)

The point of contact between a SCSI controller and a number of SCSI backplanes is faulty. The problem may lie with the SCSI cabling.

EHIOS error <*hex_value*> initializing device

The SCSI drive could not be configured.

Error:	Error:
0x01 Illegal logical channel no. (unequal 0, 1, 2, 3)	0x0c Driver got two open calls without CLOSE
0x02 Illegal HW channel no. (wrong entry in CTLR_DESC)	0x0d Wrong driver interface parameter
0x03 Wrong channel type (not SCSI or slot is free)	0x80 No error
0x04 Probe failed (read access to SCNTL0_REG failed)	0x81 SCSI target error (error bits in SCSI register dstat, sist0 or sist1)
0x05 Submodule BIST failed (BIST bit in CTLR_DESC)	0x82 Less data returned
0x06 BORIS configuration failed (SPA_ERROR occurred)	0x83 Check SCSI status error
0x07 Unknown SCSI width (code unknown)	0x84 Selection timeout
0x08 HW revision too old (submodule HW revision too old)	0x85 Operation timeout
0x09 SCSI processor error (register increment write/read test)	0x86 Operation rejected
0x0a SCSI script test execution error (INT operation)	0xA0 Illegal PCI type
0x0b Unknown controller type	0xA1 Mode selected has not been run
	0xA2 Buffer not large enough
	0xA3 Illegal request
	0xB0 Script error interrupt
	0xB1 Script polling timeout (SCSI card has no connection to SCSI bus)
	0xB2 SCSI chip detects a parity error
	0xB3 SCSI device was not connected

EHIOS init error: submodule BIST failed

A SCSI controller error has been detected in the BIST.

Error <*hex_value*> enabling func <*hex_value*> (error code <*hex_value*>)

A system warm start has been performed.

Error with eip communication
eip send buffer: <*string*>
eip receive buffer: <*string*>

An error has occurred during communication with the EIP. Possible cause: The type of system cabinet has not been entered correctly. The value *30* must be entered as the cabinet type for RM600-5xx, -6xx and -7xx systems, and *32* for RM600-2xx, -3xx and -4xx systems.

- Call the EIP command mode in the board debug monitor with *eip*.
- Look up the cabinet type:

  EIP_CMD: **rtyp**

- Correct the cabinet type if the wrong one has been defined:

  EIP_CMD: **pid**□□/* Establish physical ID of the EIP */□
  EIP_CMD: **wtyp <*pid*> <*cabinet_type*>**□□/* Enter cabinet type */

- Return to the board debug monitor:

  EIP_CMD: **.**

Function <*hex_value*> already enabled

A system warm start has been performed.

HIOS error <*hex_value*> initializing device

The SCSI drive could not be configured.

Error:	Error:
0x01□ Illegal logical channel no. □ □ (unequal 0, 1, 2, 3)□	0x0c□ Driver got two open calls without□ □ CLOSE□
0x02□ Illegal HW channel no. □ □ (wrong entry in CTLR_DESC)□	0x0d□ Wrong driver interface parameter□
0x03□ Wrong channel type □ □ (not SCSI or slot is free)□	0x80□ No error□
0x04□ Probe failed (read access to□ □ SCNTL0_REG failed)□	0x81□ SCSI target error (error bits in□ □ SCSI register dstat, sist0 or sist1)□
0x05□ Submodule BIST failed □ □ (BIST bit in CTLR_DESC)□	0x82□ Less data returned□
0x06□ BORIS configuration failed□ □ (SPA_ERROR occurred)□	0x83□ Check SCSI status error□
0x07□ Unknown SCSI width □ □ (code unknown)□	0x84□ Selection timeout□
0x08□ HW revision too old (submodule□ □ HW revision too old)□	0x85□ Operation timeout□
0x09□ SCSI processor error (register□ □ increment write/read test)□	0x86□ Operation rejected□
0x0a□ SCSI script test execution error□ □ (INT operation)□	0xA0□ Illegal PCI type□
0x0b Unknown controller type	0xA1□ Mode selected has not been run□
	0xA2□ Buffer not large enough□
	0xA3□ Illegal request□
	0xB0□ Script error interrupt□
	0xB1□ Script polling timeout (SCSI card□ □ has no connection to SCSI bus)□
	0xB2□ SCSI chip detects a parity error□
	0xB3 SCSI device was not connected

HIOS init error: submodule BIST failed

A SCSI controller error has been detected in the BIST.

Illegal bootstring

The bootstring has incorrect syntax.

Initialization of DCS board failed

The DCS controller could not be initialized.

Initialization of SCSI controller <*hex_value*> failed

The SCSI controller could not be initialized.

Controller:

0x00 CSI
0x01 DCS
0x02 HIOS or EHIOS

Invalid disk ID in NVRAM: *"<disk_id>"*

The logical boot string has been found in the root structure of the NVRAM but the entered disk ID does not comply with naming conventions, or the root structure in the NVRAM contains incorrect data.

Invalid logical boot string: *"<boot_string>"*

The logical boot string has not been found in the root structure of the NVRAM, or the boot string is incorrect, or the root structure in the NVRAM contains incorrect data.

Job error: cc <*hex_value*>, SCSI status <*hex_value*>, job status <*hex_value*>,
req sense <*hex_value*>

cc:

0x00 No error
0x01 SCSI target error (error bits in SCSI register dstat, sist0 or sist1)
0x02 Less data returned
0x03 Check SCSI status error
0x04 Selection timeout
0x05 Operation timeout
0x06 Operation rejected
x020 Illegal PCI type

cc:

0x21 Mode selected has not been run
0x22 Buffer not large enough
0x23 Illegal request
0x30 Script error interrupt
0x31 Script polling timeout (SCSI card has no connection to SCSI bus)
0x32 SCSI chip detects a parity error
0x33 SCSI device was not disconnected

SCSI status:

0x00 Target has successfully completed the command
0x02 A condition occurs that causes sense data
0x04 No error
0x08 Target is unable to accept a command from initiator
0x10 Return code for linked SCSI commands
0x14 No error
0x18 Access attempt to logical device that is reserved

job status:

0x00 SCSI job was executed with no error
0x01 A request sense command was requested
0x02 A request sense command failed
0x03 An error occurred while executing the SCSI command
0x04 SCSI job was aborted (look at cc for details)

req sense:

0x00 No sense
0x01 Recovered error
0x02 Target is not ready to accept a command
0x03 Store medium is defective
0x04 Hardware error
0x05 Target can't execute this command
0x06 Unit attention
0x07 Cartridge is write-protected
0x08 No-data condition encountered on tape

req sense:

0x09 Vendor unique
0x0a Copy command was aborted
0x0b Driver aborted the command
0x0c Search data command has satisfied on equal
0x0d Physical EOM with data still in buffer
0x0e Verify command has found an mismatch

Memory or port error on CSI

A system warm start has been performed.

More than one SCSI bus connected
EIP SCSI reset bitmap for SCSI backplane 1: <*hex_value*>
EIP SCSI reset bitmap for SCSI backplane 0: <*hex_value*>

An error has occurred in establishing the SCSI daisy chain with the aid of the EIP (RM600-5xx, -6xx, -7xx).

More than one SCSI bus connected, EIP SCSI reset bitmap: <*hex_value*>

An error has occurred in establishing the SCSI daisy chain with the aid of the EIP (RM600-2xx, -3xx, -4xx).

mt(<*scsi_bus,scsi_id*8*>,0): specified device not a streamer tape

The selected drive is an 8mm MTC drive.

No disk with disk ID *"<disk_id>"*

The logical boot string has been found in the root structure of the NVRAM but the associated physical boot string is invalid. The entered disk ID does not match any disk inquiry string in the system, or the root structure in the NVRAM contains incorrect data.

No ELF file

The magic number in the ELF header of the program to be booted is incorrect. Either the wrong removable medium has been chosen, or the disk is not a system disk.

No ELF header

e_phnum in the header is not *2*. The found program is not suitable for booting. Either it is the wrong file on the removable medium or the disk is not a system disk.

No Ethernet controller available

No Ethernet controller has been found in the configuration table.

No HIOS or EHIOS board on SP bus

No HIOS or EHIOS board has been found in the configuration table. Either no such board has been installed in the system or the board is defective.

No LAN controller available

No LAN controller card has been found on the HIOS or EHIOS board. Either no LAN card has been inserted or the board or card is defective.

No SCSI channel on <*string*> board

No SCSI controller card has been found on the HIOS or EHIOS board. Either no card has been inserted or the board or card is defective.

NVRAM not initialized, error: <*hex_value*>

The NVRAM of the console controller is defective or the battery is flat.

Error:		Error:	
Bit 0:	SRAM corrupted by power failure	Bit 2:	Invalid address area
Bit 1:	Data missing	Bit 3:	Modifier unknown

Reached statement after return_to_firmware

The CPU firmware has returned control to boot1. There is a hardware or software error.

SCSI backplane has type F, can't get SCSI configuration in T5 system
EIP send buffer: <*string*>
EIP receive buffer: <*string*>

In RM600 E systems, boot1 first establishes a SCSI daisy chain configuration for the system cabinet. This is not possible with the SCSI backplane of type F.

SCSI chip register: DSTAT=<*hex_value*>, ISTAT=<*hex_value*>, SIST0=<*hex_value*>,□
SIST1=<*hex_value*>, Irupt<*#*>=<*hex_value*>

Error details of the SCSI driver for the HIOS or EHIOS board.

SCSI_TEST failed

The SCSI command *test unit ready* was not executed correctly.

SD[<*scsi_bus,scsi_id***8*>]: inquiry command failed

The SCSI command *do inquiry* was not executed correctly.

SD[<*scsi_bus,scsi_id***8*>]: invalid block 0 on disk

The partition information in block 0 of the disk is invalid.

SD[<*scsi_bus,scsi_id***8*>]: invalid command <*hex_value*> in strategy routine

An invalid SCSI command has occurred in the SCSI driver.

SD[<*scsi_bus,scsi_id***8*>]: invalid partition info on disk

The partition information in block 0 of the disk is invalid.

SD[<*scsi_bus,scsi_id***8*>]: IO error on reading partition info

The partition information in block 0 of the disk could not be read.

SD[<*scsi_bus,scsi_id***8*>]: mode select failed

The SCSI command *mode select* was not executed correctly.

SD[<*scsi_bus,scsi_id***8*>]: mode sense failed

The SCSI command *mode sense* was not executed correctly.

SD[<*scsi_bus,scsi_id***8*>]: read capacity failed

The SCSI command *read capacity* was not executed correctly.

SD[<*scsi_bus,scsi_id***8*>]: read capacity: block size 0

The SCSI command *read capacity* has returned block size *0*.

SD[<*scsi_bus,scsi_id***8*>]: request sense failed

The SCSI command *request sense* was not executed correctly.

SD[<*scsi_bus,scsi_id*8*>] SCSI command <*hex_value*> failed

The specified SCSI command was not executed.

SCSI command:

0x00 Test unit ready
0x01 Rezero unit
0x03 Request sense
0x04 Format unit
0x08 Read
0x0a Write
0x0b Seek
0x0f Translate logical to physical
0x11 Space
0x12 Do inquiry
0x13 Write buffer
0x14 Read buffer

SCSI command:

0x15 Mode select
0x16 Reserve unit
0x17 Release unit
0x1a Mode sense
0x1b Start/stop unit
0x1c Receive diagnostic
0x1d Send diagnostic
0x22 Do long inquiry
0x25 Read capacity
0x28 Read extended
0x2a Write extended

SD[<*scsi_bus,scsi_id*8*>]: start unit failed

The SCSI command *start unit* was not executed correctly.

SD[<*scsi_bus,scsi_id*8*>]: test unit failed

The SCSI command *test unit* was not executed correctly.

SPA_E_ERROR closing device

The SCSI driver for the HIOS or EHIOS board detected an error on closing the SCSI connection to a drive.

ST(<*scsi_bus,scsi_id*8*>,0): device type <*hex_value*> !=0x01

The selected drive is not a tape drive.

Device type:

0x00 Direct access
0x01 Sequential access
0x7F Logical unit not present

ST(<*scsi_bus*,*scsi_id***8*>,0): hard error <*hex_value*> on SCSI command

An error has occurred with a SCSI command.

Error:		Error:	
0x00	No sense	0x08	No-data condition encountered on
0x01	Recovered error	0x09	tape
0x02	The target is not ready to accept a command	0x0a	Vendor unique
0x03		0x0b	Copy command was aborted
0x04	The storage medium is defective	0x0c	Drive aborted the command
0x05	Hardware error		Search_data command has satisfied on equal
	The target can't execute this command	0x0d	
0x06			Physical EOM with data still in buffer
0x07	Unit attention	0x0e	
	Cartridge is write-protected		Verify command has found an mismatch

ST(<*scsi_bus*,*scsi_id***8*>,0): init of SCSI controller failed

The SCSI controller could not be initialized.

ST(<*scsi_bus*,*scsi_id***8*>,0): inquiry command failed

The SCSI command *do inquiry* was not executed correctly.

ST(<*scsi_bus*,*scsi_id***8*>,0): mode select failed

The SCSI command *mode select* was not executed correctly.

ST(<*scsi_bus*,*scsi_id***8*>,0): request sense failed

The SCSI command *request sense* was not executed correctly.

ST(<*scsi_bus*,*scsi_id***8*>,0): rewind of tape failed

The tape could not be rewound.

ST(<*scsi_bus*,*scsi_id***8*>,0): space to requested file failed

The SCSI command *space* was not executed correctly.

ST(<*scsi_bus*,*scsi_id***8*>,0): tape is write protected

The tape is write-protected.

ST(<*scsi_bus*,*scsi_id***8*>,0): test unit failed

The SCSI command *test unit* was not executed correctly.

ST(<*scsi_bus*,*scsi_id***8*>,0): transfer lost more date than requested

Transfer of data from tape did not work properly.

ST(<*scsi_bus*,*scsi_id***8*>,0): unknown device name

The device name specified in the boot string is not known.

Allowed:		Allowed:	
cd□	CD-ROM□	mt, st□	¼" MTC□
sd, hd□	SCSI disk□	vt□	8mm MTC□
dt□	DAT□	et	Ethernet
fd	Floppy		

st_space: illegal code for space command <*hex_value*>

The SCSI command *space* was not executed correctly.

st_strategy: illegal read/write command

Error in the SCSI driver.

st_strategy: transfer count <*hex_value*> no multiple of block size <*hex_value*>

Error in the SCSI driver.

Test stat flag of IO controller is set

The test status flag of the SCSI controller is set. The SCSI controller is defective.

Timeout waiting for command end

Timeout with a command for the MBP.

Too many devices open

Error in the SCSI driver.

Type error in regtable

Error in the SCSI driver.

Undefined LAN channel id <*dec_value*>.□
Use 32...63 for "X86 LAN channel id"

An incorrect LAN channel ID was specified (first parameter in parentheses).

Undefined channel id <*dec_value*>.□
Use □0... 5 for "logical SCSI bus id"□
Use 32...63 for "logical X86 SCSI channel id"□
Use 64...95 for "logical PCI SCSI channel id"

An incorrect channel ID was specified (first parameter in parentheses).

Undefined SCSI operation

Error in the SCSI driver.

Unexpected SCSI status <*hex_value*>

An unexpected SCSI status exists.

Status:

0x00 Target has successfully completed the command
0x02 A condition occurs that causes sense data
0x04 No error
0x08 Target is unable to accept a command from initiator
0x10 Return code for linked SCSI commands
0x14 No error
0x18 Access attempt to logical device that is reserved

Unknown system type in config table: <*dec_value*>
(0: R4-MR, 1: R4-HE, 3: T5-MR, 4: T5-HE)

The firmware of the console controller has entered an incorrect system type in the configuration table. Possible cause: The type of system cabinet has not been entered correctly. The value *30* must be entered as the cabinet type for RM600-5xx, -6xx, -7xx and -E60 systems, and the value *32* for RM600-2xx, -3xx, -4xx and -E20 systems.

- Call the EIP command mode in the board debug monitor with *eip*.
- Look up the cabinet type:

  EIP_CMD: **rtyp**

- Correct the cabinet type if the wrong one has been defined:

  EIP_CMD: **pid** /* Establish physical ID of the EIP */
  EIP_CMD: **wtyp <*pid*> <*cabinet_type*>** /* Enter cabinet type */

- Return to the board debug monitor:

  EIP_CMD: **.**

vt(<*scsi_bus,scsi_id*8*>,0): specified device not a video8-tape

The selected drive is not an 8mm MTC drive.

Warning, target was not disconnected

The drive has not been removed from the SCSI connection.

Wrong pointer to root structure in NVRAM

The NVRAM of the console controller has not been initialized. There is no pointer to the root structure in the NVRAM.

Wrong root magic in NVRAM

The NVRAM of the console controller has not been initialized. The wrong magic number is in the root structure in the NVRAM.

You have chosen ethernet channel <*dec_value*>, no such channel available

The Ethernet channel in the boot string (first parameter in parentheses) is too high.

You have chosen LAN channel <*dec_value*>, no such channel available

The LAN channel in the boot string (first parameter in parentheses) is too high.

You want to log SCSI bus <*dec_value*>, no such SCSI bus connected to the system

The selected logical SCSI bus in the SCSI bus boot string (first parameter in parentheses) is not connected to the system.

You want to log PCI SCSI chan id <*dec_value*>, no such SCSI channel on <*string*> board

The channel ID in the bootstring is too high. The controller configuration table will also be displayed.

You want to log SCSI chan id <*dec_value*>, no such SCSI channel on <*string*> board

The channel ID in the bootstring is too high. The controller configuration table will also be displayed.

You want to log X86 SCSI chan id <*dec_value*>, no such SCSI channel on <*string*> board

The channel ID in the bootstring is too high. The controller configuration table will also be displayed.

17 Error messages relating to system calls

A failed system call returns a return value other than *0* to the calling process. In most cases this will be the value *-1*. An error number is also returned in the external variable *errno*. The variable *errno* is not reset after successful system calls so it should only be queried after an error.

All error numbers and error messages defined in the file *errno.h* are described in the *intro_prm2*(2) manual page.

Abbreviations

AFP	Alternating edge pulse transmission method
BBU	Battery Backup Unit
BIST	Built-In Self-Test
BNC	Bajonet Norm Connector
BNU	Basic Networking Utilities
BORIS	Buffer-Oriented Remote Interface for SP-Bus
CAN	Controller Area Network
CD-ROM	Compact Disk - Read Only Memory
CPU	Central Processing Unit
CSI	Central Services & Interfaces
CTRL	Control key. This key is labelled [Strg] (Steuerung) on German keyboards.
DAT	Digital Audio Tape
DC	Direct Current
DCS	Disk Controller - SCSI
DCU	Device Control Unit
DIP	Dual Inline Package
DSSI	Digital Storage Systems Interconnect
EDEC	Error Detection and Error Correction
EEC	Enhanced Enclosure Control
EEPROM	Electrically Erasable Programmable Read-Only Memory
EHIOS	Extended High-Performance Input/Output System
EIP	Environmental Interface Processor

ELF	Extensible Linking Format
EMM	Error Message Management
FMLI	Form and Menu Language Interpreter
HIOS	High-Performance Input/Output System
IDC	ISA Diagnostic Bus Controller
IHSS	Inhouse interface
IKDB	Internal Kernel Debugger
IMD	Inhouse Multiplexer Data Transmission Unit
IPC	Interprocess Communication
ISA	Industry Standard Architecture
LAR	Lockout Auto Recovery
LCD	Liquid Crystal Display
LED	Light Emitting Diode
MB	Multi Bus
MBP	Multi Bus Processor
MTC	Magnetic Tape Cartridge
NFS	Network File System
NVRAM	Non-Volatile Random Access Memory
OLR	Online Replacement
PCI	Peripheral Component Interconnect
SAC	Service Access Controller
SASH	Stand Alone Shell
SCSI	Small Computer Systems Interface

SNA	System Network Architecture
SPA	Synchronous Pipelined Architecture
SP-bus	Synchonous Pipelined Bus
SRAM	Static Random Access Memory
TAK	Terminal attachment concentrator
TCP/IP	Transmission Control Protocol/Internet Protocol
TDS	Test and Diagnostic System
TED	Triteal Desktop
TTY	TeleType
UBFT	UPS BBU Fan Temperature Control
UPS	Uninterruptible Power Supply
UUCP	UNIX to UNIX Copy
VGA	Video Graphics Adapter

Related publications

Please apply to your local office for ordering the manuals.

DSM Event Reporting
Monitoring Events in Networks
Product Manual

Target Group
System administrators

Contents
The manual is part of a series of product manuals describing the administration and control of computer systems in networks. This manual describes how messages about events (i.e. exceptional states and error situations) can be received on systems in a network and be passed on to a higher-level system on the basis of certain criteria.

Online Test System
TDS2
Product Manual

Target Group
Service, system administrators

Contents
This manual describes the test system and diagnostic system, TDS2, which is used to test hardware components.

Reliant UNIX 5.45
Commands. User's Reference Manual

Target Group
Users, system administrators

Contents
The manual describes the user commands of the Reliant UNIX 5.44 operating system.

Reliant UNIX 5.44
Hardware Configuration with Config under SINIX/windows
Product Manual

Target Group
System administrators

Contents
The manual describes the functionality of Config to configure RM systems using the graphical SINIX/windows user interface.

Reliant UNIX 5.45
Networking Reference Manual

Target Group
Programmers, system administrators, users

Contents
The manual describes commands, functions, utilities and system files needed to work with computers which communicate in a network.

Reliant UNIX 5.45
Programmer's Reference Manual

Target Group
Programmers

Contents
The manual describes commands, system calls, library functions, file formats and tools for developing C programs on computers running under the Reliant UNIX 5.44 operating system.

Reliant UNIX 5.45
Reliant UNIX Installation
Installation Guide

Target Group
System administrators

Contents
The manual describes the re-installation and updating of Reliant UNIX as well as initial installation of it when there is a new system disk.

Reliant UNIX 5.45
Reliant UNIX Operation
System Administrator's Guide

Target Group
System administrators

Contents
This manual describes the administration of Reliant UNIX.

Reliant UNIX 5.44
System Administration and Hardware Configuration Using the SYSADM Interface
System Administrator's Guide

Target Group
System administrators

Contents
The manual describes the administration and configuration of RM systems alphanumeric SYSADM user interface.

Reliant UNIX 5.45
System Administration within a Domain
Systemverwalterhandbuch

Target Group
System administrators

Contents
This manual describes how several nodes can be bundled in an administration domain and thus managed jointly. This is possible using a domain administration system, in this case the DomainAdmin add-on product. Users are given access to the essential functions of this administration system by means of the WebSysAdmin (WSA = Web based System Administration) user interface. This manual deals less with DomainAdmin itself, than with using DomainAdmin via the WSA user interface for managing the systems in an administration domain.

Reliant UNIX 5.44
System Administrator's Guide
User Guide

Target Group
System administrators

Contents
The manual provides an introduction to UNIX system administration. It describes, amongst other things, system administration tools, the configuration and support of UNIX systems, system accounting and aspects of system security. The manual is valid for both MIPS-based and Intel-based UNIX systems.

Reliant UNIX 5.45
System Administrator's Reference Manual

Target Group
System administrators

Contents
The manual describes commands and application programs for system maintenance as well as file formats and special files for administration of the Reliant UNIX 5.44 operating system.

Reliant UNIX 5.45
Tuning Guide
System Administrator's Guide

Target Group
System administrators

Contents
The manual describes how the system performance can be analyzed and improved (tuning), and when the hardware should be extended. The manual contains also a description of the kernel parameters.

Reliant UNIX 5.44
Virtual Disks
System Administrator's Guide

Target Group
System administrators

Contents
The manual describes the various types of virtual disks and how to configure them.

Reliant UNIX 5.45
Configuring Virtual Disks using VDisk Lite
System Administrator's Guide

Target Group
System administrators

Contents
This manual describes how to configure virtual disks using VDisk Lite. VDisk Lite provides a convenient user interface for configuring virtual disks. It manages disk areas and guards against errors in the configuration.

ServerView
Server Management
User's Guide

Target Group
System administrators, network administrators, service

Contents
This manual gives you an overview of the basics of server monitoring, documents the operating system-specific requirements for the use of ServerView, and describes the process of installing ServerView and its user interface.

SINIX 5.41
STREAMS
Programmer's Guide

Target Group
System developers, application programmers designing driver and communication software

Contents
The manual describes the UNIX System V STREAMS mechanism. It provides a general overview, deals with the individual components and supplies programming guidelines. It also includes a reference section.

TransView Control Center (UNIX)
TransView Control Center Agent (UNIX, Windows NT)
Enterprise Management for Client/Server Environments
User's Guide

Target Group
Network planners, network administrators and network operators

Contents
The TRANSVIEW Control Center integrates a wide range of applications for system, application and network management. The manual describes all the functions of the Control Center and the Agent (graphical user interface and commands).

Xprint (Reliant UNIX])
User and Administrator Guide

Target Group
Users and administrators of Xprint

Contents
The manual describes the methods used for working with Xprint and for its administration and installation.