



## SINIX/windows *ONLINE Documentation*

# SINIX/windows User Environment V3.0 Guide for Experts and System Administrators

Edition April 1996

---

SINIX® Copyright © Siemens Nixdorf Informationssysteme AG 1990.  
SINIX is the UNIX® System derivative of Siemens Nixdorf Informationssysteme AG.  
UNIX is a registered trademark in the United States and other countries,  
licensed exclusively through X/Open/x11Company/x11Limited.  
Base: OSF/Motif(TM), Copyright © Open Software Foundation, Inc.  
X Window System(TM), Copyright © Massachusetts Institute of Technology.  
OSF/Motif is a registered trademark of Open Software Foundation, Inc.  
X Window System is a registered trademark of Massachusetts Institute of Technology.  
Copyright © Siemens Nixdorf Informationssysteme AG 1996.  
All rights reserved.  
Delivery subject to availability; right of technical modifications reserved.  
All hardware and software names used are trademarks of their respective manufacturers.  
**Siemens Nixdorf Informationssysteme AG**

---

# 1 Preface

The SINIX®/windows User Environment is an OSF/Motif-based user interface for SINIX systems running SINIX V5.41 and above. It is a graphical interface with which you work interactively.

It is an integrated, consistent working environment in which you can access applications on your screen using a mouse and the keyboard. The SINIX/windows User Environment owes its consistency of appearance and behavior to the fact that it was developed on the basis of OSF/Motif™, which itself builds on the network-transparent, hardware-independent X Window System™.

## 1.1 Target group and manual structure

The present manual, "SINIX/windows User Environment; Guide for Experts and System Administrators", is intended for experienced users and system administrators of the Motif-based user interface on SINIX systems. It builds on the knowledge base provided by the manual "SINIX/windows User Environment; Introduction to Handling and Configuration". It gives an account of the client-server architecture of the X Window System; it discusses how the X server is used to configure color and font selections; and it explains the resource mechanism of the X Window System with a view to enabling experienced users to customize X applications to suit their own preferences. Experts and system administrators are introduced to the xdm display manager's session configuration capabilities and to menu definition using the mwm window manager. They also receive an overview of the the system management tools which SINIX/windows provides.

## 1.2 Notational conventions

This guide uses the following conventions:

Example	Meaning
"Window"	Cross-references to chapters and sections in this manual and cross-references to other manuals are placed in double quotation marks.
"Editor"	In running text, a string of characters that you see on the screen is placed in single quotation marks.
\$PATH	This font is used for screen output in examples or for entries that you must make in the exact same format.
<i>host_name</i>	Italics are used in commands to describe the sort of information that you type, rather than the specific characters.
[name ]	A term in square brackets stands for optional input.

Table 1: Notational conventions

### 1.3 Changes since the last version of the manual

The tables below list the functional changes made to clients between Version 2.2 and Version 3.0 of the SINIX/windows User Environment. Changes of an editorial nature and corrections of mistakes in the documentation are only listed if they are of particular significance.

Topic	Brief description of change
New Desktools	A number of new desktools are available: Desktools that are relevant for the system administrator include: 'Log-Browser', 'System Information'. The "Help" item in the pop-up menu of each desktool provides detailed information on the functions and operations that you can use to complete the specified task.

Table 2: Changes between Version 2.2 and Version 3.0

## 2 Principles of SINIX/windows operation

SINIX/windows operates on the client-server principle of the X Window System. The "Clients, X servers and displays" section of this chapter discusses the components of the X Window System in terms of the client-server model and explains what they do. The section headed "Communication between client and X server" explains further aspects of the client-server model. The preconditions for networked communication between clients and the X server are dealt with in the section on "Communication across a network". Information on related topics is given in the sections headed "Controlling client access to the X server".

### 2.1 Clients, X servers and displays

The architecture of the X Window System is based on the client-server model, in which a distinction is drawn between two groups of programs, referred to as X clients and X servers. Owing to the two-part architecture of the X Window System, clients and X servers can either be on the same host or on different hosts linked by a network.

Clients are what you use to perform work on your workspace. A client in X Window System terminology is an application which exchanges data with one or more X servers. It is known as a client because it is provided with input and output services by the X server. Clients call on the X server to carry out actions for them. They have no direct influence on a window or screen; they can only ask the X server to do something for them, such as move a window, color an element or write text in a window. The clients are thus largely independent of the input/output unit hardware. Examples of clients include mwm, xdesk and xdm.

X servers offer services to clients. An X server is an application which converts requests from a client into output on the display screen and passes user input from the keyboard or a pointing device to the designated client. Thus the X server acts as an intermediary between a client or clients and the input/output unit (the display) that it controls. It uses a hardware-independent interface to communicate with clients, and hardware-dependent interfaces to communicate with the components of the display.

An X server is assigned to a single display. A display consists of one or more logical screens, a keyboard and a graphical pointing device such as a mouse. A display is generally associated with just one logical screen, which in hardware terms is simply a monitor. However, the X Window System architecture does also allow a monitor to offer multiple logical screens and even permits the screens of multiple monitors to form a single display. The X server does not reveal to the clients how many monitors are represented by the logical screens; it simply makes the logical screens accessible.

Clients read the DISPLAY environment variable to find out which X server to address and which screen to pass output to via the X server. Alternatively, you can specify a different display in the -display option when you call a client.

A display is defined by the name of the host to which it is connected, the display number and the screen number, using the following syntax:

```
[host-name]:display-number[.screen-number]
```

The display number has to be specified because it is possible to start a number of servers for different displays on a single host. Display numbers are assigned from 0 upwards.

The screen number can be specified if a display consists of more than one screen. Screen

numbers, too, are assigned from the default 0 upwards.

Dividing the X Window System architecture into two components enables clients and X servers to run either on the same host or on different hosts which are interconnected via a network.

The client/server separation is typically exploited by X terminals. In their role as terminals, X terminals provide a display for data input and output. At the same time X terminals are also self-contained systems in the network on which an X server for the display is running. They give you access via the X server to the clients on the hosts in the network at which you work.

Hosts which do not have a graphics screen can thus use any available display in the network for their graphics output. Processes which make intensive use of CPU time can be transferred from workstations which have a limited processing capability to faster machines; they can, however, be controlled from the workstation's high quality graphics screen using a convenient graphical user interface.

## 2.2 Communication between client and X server

The X server and the client communicate through a standardized, hardware-independent interface, the X Protocol.

When you call a client, the client sets up a connection to the X server and sends it a request, which might be to display a window on a screen. The X server in this case then has the window displayed on a screen that it controls. The X server also colors the window in accordance with the requests from the client and/or the specifications in its configuration files. It likewise matches the requests to the capabilities of the monitor and the graphics card. It displays text in the fonts it has loaded, again allowing for the capabilities of the monitor and the graphics card. This is why it is always uniquely associated with a single display.

The X server also passes user input to the appropriate clients. For example, if you position the mouse pointer on a button in a client window and press the <BSelect> mouse button, the X server registers this event and tells the client about it. The client then tells the X server how to react to the event, and the X server responds by generating suitable output on the screen which it controls. Depending on the nature of the request from the client, the X server may inform the client of any errors which occurred while it was handling input and output for the client.

The X server does nothing on its own initiative. Its function is to service requests from clients and to react to the actions taken by the user at the input devices.

An X server can handle the graphical presentation and output requirements of a number of clients simultaneously. Similarly, a user can interact with a number of clients on the screen; but user input can at any given moment only be directed at one client, or at one client control component.

The X server allocates user input to whichever window or graphical control is active at the time. The active window or control is thus the one which currently has the "input focus". The input focus is assigned either implicitly to the window in which the pointer is positioned or explicitly to the window in which the user clicks a mouse button. The mwm KeyboardFocusPolicy resource defines which of these input focus assignment policies applies. User input cannot be directed to client windows which do not have the input focus.

## 2.3 Communication across a network

If an X server and its clients are on different computers in a network, they use both the X Protocol and the network's transport protocol for message passing. SINIX/windows supports the TCP transport protocol from the TCP/IP protocol family. The DISPLAY environment variable which all clients evaluate controls connection setup on the client side.

The transport protocol is responsible for establishing a connection between two communication entities on different hosts in the network and for securely transmitting data over this connection. For the two communication entities to be able to exchange information, they have to be using the same transport protocol.

A host required to communicate using transport protocols must obviously have a fully functioning set of network facilities. This means that:

- the physical link must be installed.
- the host names must be known. For TCP/IP the network addresses and host names are defined in the /etc/hosts file or are looked up in the NIS (Network Information System).
- the network services must be accessible. For TCP/IP the association between the X server and the transport protocol port is defined in the /etc/services file or is looked up in the NIS (Network Information System).

These conditions must be created at the time of network software installation.

## 2.4 Controlling client access to the X server

There are two approaches to grant or restrict client access to an X server. Using the `xhost` client you can define host-specific restrictions; with a user-specific authorization mechanism you can control client access to the X server within a user session.

### 2.4.1 `xhost` - host-specific access control

The `xhost` client protects the X server against unauthorized from hosts on the network. As a security precaution, the X server uses an internal access control list to determine whether to accept connections to clients from a particular host. When the X server is booted, its internal access control list is initialized from an X server configuration file, as described in the documentation for the X server you use for your display. You can use the `xhost` client to change this internal list, so as to grant access to other hosts or remove access authorization from listed hosts. Any changes you make apply only for the duration of the session; the next time the X server is started, the access control list will be reinitialized. You can use the `xhost` client to specify the hosts from which clients can access the X server or to display a list of the machines that have access to the X server.

Entering the command below on the local host adds the remote host named *hostname* to the X server's internal list.

```
xhost +hostname
```

Thereafter clients running on host *hostname* will be able to communicate with the X server. Note that the `xhost` client, like every other client, reads the `DISPLAY` environment variable to find out which X server to use. If the authorization mechanism is active for the X server, the `xhost` client must also be able to supply the session key (see section "Authorization - user-specific access control").

### 2.4.2 Authorization - user-specific access control

Instead of the host-specific access control provided by the `xhost` client, it is also possible to implement user-specific access control. User-specific access control is handled by an authorization mechanism which checks whether the client is authorized to access the X server in the course of a session. This authorization mechanism is supplied by the `xdm` display manager.

If the authorization mechanism is active for the local display, then when the local X server is started the `xdm` display manager generates a session key and stores it in a file. On startup the X server is told the name of this file. From then on the local X server grants access only to X clients which can present the session key.

Remote X servers using the X Display Manager Control Protocol (XDMCP), such as X servers for X terminals, are supplied with the session key by `xdm` via the XDMCP networking protocol if they request `xdm` services and ask for authorization to be used. If a user has successfully logged in at the `xdm` login window and authorization is active, the display manager also stores the session key in the file identified by `$XAUTHORITY`. If the `XAUTHORITY` environment variable is not set, the default file is `.Xauthority` in the user's home directory. Any clients the user starts up within the session for the associated display retrieve the session key from the authorization file and present it to the X server. The X server rejects other users' X clients if they cannot supply the current key. Permission to access the authorization file should be restricted to the owner. This ensures that it is impossible for unauthorized users to run a client on the X server.

When the session is terminated and a new one is started, a new key is generated and communicated to the X server.

#### 2.4.2.1 Activating the authorization mechanism

To activate the authorization mechanism, add the following resource definitions to the xdm display manager configuration file (usually xdm-config):

```
DisplayManager*authorize: true
```

This entry specifies that authorization is to be activated for all connections. Authorization can be deactivated for local displays with the entry

```
DisplayManager._0.authorize: false
```

With remote X servers which use XDMCP, the setting (True or False) of xdm's authorize resource does not govern whether or not authorization is active. In this case if an X server asks for authorization to be used when it requests xdm services, xdm will always provide authorization for that X server.

If message indicating that the session is not secure appears in the X terminal's xdm login window, it means that the display manager has specified that the authorization mechanism should be used for the X terminal, but that the X terminal's X server does not support authorization and that authorization has not been activated.

#### 2.4.2.2 Authorization for clients on remote hosts

If clients which access the display need to be run on other hosts in the network, the session key must be made available to them. This entails extracting the session key from the local authorization file and adding it to the user-specific authorization files on the remote hosts.

The tools provided for this purpose are the `xauth` application, the `xauth_remote` shell script and the "Authorize" menu item on the pop-up menu of a desktool.

If a user with the login name "custer" wants to extract the authorization entry for the display "asterix:0.0" and transfer it to his authorization file on the host "distant", he enters the following:

```
xauth extract - asterix:0.0 | rsh distant /usr/bin/X11/xauth merge -
```

The command `xauth extract` extracts the authorization entry for the display "asterix:0.0" from the file identified by `$XAUTHORITY`, or from `$HOME/.Xauthority`, and writes it to standard output (-), which is passed to the host "distant" by the `rsh` command. On "distant" the entry is added to the authorization file by `xauth merge`.

The command

```
xauth_remote distant
```

has the same effect. This shell script also allows for the fact that the entries for a local display in the authorization file look different from those for remote displays and so need to be converted.

#### 2.4.3 Interaction between host-specific and user-specific access control

The host-specific access authorization granted by the `xhost` client takes precedence over the user-specific access authorization provided by the authorization mechanism. If a host is granted access to a server by the X server's internal host list, all clients started on that host are authorized to access that server, even if they cannot supply the valid session key. Hosts on which user-specific authorization is to apply must not appear in the X server's internal host-specific access control list.



## 3 Graphical presentation

The quality and quantity of the colors that can be displayed are governed by the capabilities of the monitor, by the installed graphics card and by the visual class supported by the X server. Font presentation, too, is influenced by the resolution of the monitor and the graphics card. The X server is responsible for supplying the colors and fonts, and conceals these hardware details from the clients.

The clients request the X server to use certain colors and fonts for their outputs. You can therefore determine the availability of colors and fonts when you configure the X server. Which fonts you can use depends on which fonts are available in the directories on the X server's font path. In supplying fonts the X server can use the services of a font server, provided that the font server can be reached via the X server's font path and the font server is running on one of the hosts in the network.

You can specify the colors and fonts which the client is to request from the X server, using the client's resources. Typical resource classes for colors are Foreground and Background; typical resource classes for fonts are Font and FontList. The X server can set up an internal database for resources and their values in response to an explicit request to do so from a client. Then when a client is called, the requested resources are added to the database.

The client can also ask the X server to tell it the current values of resources.

Usually a client will copy the resource database from the X server when invoked and then use this copy. Then requests for resources can be handled quickly without unnecessary network traffic; but the resource values will be those which applied at the time the copy was made.

Using the following command you can query the database in the X server with the `xrdb` client and write the information to standard output. The client accesses the X server via the `DISPLAY` environment variable.

```
xrdb -query
```

### 3.1 Colors

An image on the monitor is made up of an array of picture elements (pixels) arranged in lines which are scanned at a fixed frequency. The number of pixels per line and column and the image refresh rate are governed by the graphics card. With some graphics cards you can use X server options to modify the number of pixels, thereby altering the resolution. For each pixel, the color in which the monitor is to display that pixel is stored on the graphics card.

This section describes how colors are defined in the X Window System. The section headed "Color names and numeric color descriptions" is an introduction to color naming in color spaces. With clients color definitions you can choose whether to define colors in device-independent CIE color spaces or in the device-dependent RGM color space. The RGB color space is the underlying model for the X server, as the X server uses the RGB color space model to define and interpret colors in the following cases:

- when exchanging color information with clients
- when interpreting visual class and colormap entries for converting color information to actual colors displayed on the monitor.

### 3.1.1 Color names and numeric color descriptions

A color can be identified by either a color name or a numeric color description. Color names such as *dark green* and *light gray* give the user an idea of the color represented on the screen. Color names are assigned to color values in databases. The file `/usr/lib/X11/rgb.txt` generally contains (in readable form) the database of a local X server for color values in the RGB color space.

Numeric color representations identify the attributes of colors for a certain color space. A color space is a model in which colors are represented by a set of numeric values. Color spaces are supported by X11R5 and above. There are color spaces which are device-dependent and color spaces which are device-independent.

The values of a device-independent color space represent the general attributes of a color, which is then displayed according to the limitations of the graphics card and monitor. The hardware specifications must be stored on the X server, so that they can be queried by a client. The client converts the numeric color representation in accordance with the limitations of the display and requests the X server to display this color value. CIE color spaces are examples of device-independent color spaces.

The values of a device-dependent color space can be converted directly for display on the screen. Depending on the monitor and graphics card used, however, the color specified in this way may look different on the screen. The RGB color space is an example of a device-dependent color space.

### 3.1.2 CIE color spaces

CIE color spaces are device-independent color spaces. CIE stands for Commission Internationale de L'Eclairage. CIE color space are illustrated in this section with reference to the CIELab color space.

In a CIELab color space a color is described by three parameters:

- first, its brightness or luminance  $L$ , with values between 0 (dark) and 100 (bright),
- second, its red-green intensity  $a$ , with values between 1 (red) and -1 (green),
- third its blue-yellow intensity  $b$ , with values between 1 (blue) and -1 (yellow).

In the square below, red-green intensity "a" and blue-yellow intensity "b" uniquely identify a shade in CIELab color space syntax:

The valid format in the X Window system for a numeric color description in CIELab color space syntax is:

CIELab: $L/a/b$

Example: CIELab:50/-1/0 for green with 50% luminance.

### 3.1.3 RGB color space

The RGB color space is an example of a device-dependent color space. Under the X Protocol, the client and the X server mutually define colors in RGB syntax.

In the RGB color space, every pixel on the screen is made up of a mix of three colors, red, green and blue, in varying percentages. Each color component is generated by a separate electron gun. The intensities of the color component can be varied. If all three colors are at 100% intensity, the human eye perceives the pixel as white. If none of the three colors is present, the pixel appears black. Varying the proportions produces the various available shades and colors.

The client can code the intensity of each color component as 4, 8, 12 or 16 bits. The

number of bits used determines how many different shades can be generated from the various combinations of red, green and blue intensities. The X server's conversion of the codes into color is limited by the display hardware.

Number of bits per color	Number of achievable shades	Representation in the RGB color space <sup>1</sup>
4	$2^4 \times 2^4 \times 2^4$	rgb: <i>r/g/b</i>
8	$2^8 \times 2^8 \times 2^8$	rgb: <i>rr/gg/bb</i>
12	$2^{12} \times 2^{12} \times 2^{12}$	rgb: <i>rrr/ggg/bbb</i>
16	$2^{16} \times 2^{16} \times 2^{16}$	rgb: <i>rrrr/gggg/bbbb</i>
1) Color description in RGB notation valid in the X Window System		

Table 3: Relationship between number of bits per color component, number of achievable shades, and representation in the RGB color space

The letters in the RGB color space representations stand for the red (*r*), green (*g*) and blue (*b*) components. Each letter stands for a hexadecimal digit defining the intensity of the associated color component.

Color component intensities are generally defined using 4 or 8-bit values. 4-bit values allow 4096 shades to be produced, while 8-bit values allow for about 16.7 million. Thus even 8-bit RGB values can generate a huge range of different shades.

A client requests the X server to provide it with the colors it needs to draw its window. These colors are defined either in the application or in the arguments supplied by the user when calling the client or in resource description files.

### 3.1.4 Colormaps

A distinction must be drawn between hardware and software colormaps. A hardware colormap (also known as a lookup table) is a register on the graphics card. This register is dynamically updated from the software colormap generated by the X server on initialization. The term colormap as used below refers to the software colormap unless otherwise specified.

A colormap consists of an array of color cells, each containing one entry, an RGB value. The number of color cells is  $2^n$ , where  $n$  is the number of bits the graphics card allocates to each pixel. The number of bits available for a pixel is equivalent to the number of bit planes and is governed by the amount of memory on the graphics card. The color cells are numbered from 0 through  $(2^n-1)$ . This numbering system acts as an index for the colormap. The number of colors that can be displayed on the screen at any one time is a function of the size of the colormap. If there are eight bits available to represent a pixel, up to 256 ( $2^8$ ) colors can appear on the screen at any one time.

A colormap is generally used concurrently by a number of clients.

As a rule, when the X server is reset or initialized, the only colors it preallocates are black and white. As soon as a client is called, the color cells are successively assigned the RGB values the client requests from the X server. Once all available color cells have been assigned values, no more colors can be requested. Instead, the X server selects the RGB value closest to the required value from the full colormap.

Color cells can be shared between multiple clients to reduce the risk of exceeding the capacity of the colormap. Shared color cells can be used by any number of clients; but once they have been assigned values, they can no longer be changed.

It is also possible to specify a private colormap in the call to a client. This predefined private colormap overrides the graphics card's colormap when the client gets the input focus. The coloring of this client, and of its inactive surroundings, is defined on the basis of the private colormap. However, the overridden colormap is retained in the X server, and the colormap focus is reallocated to it (in other words, the original colormap is reinstated) when the client loses the input focus. If the current colormap differs from the private one, the colors of the client with the private colormap will appear as false colors, as coloring for all clients is controlled by the current hardware colormap only.

### 3.1.5 Visual classes

The default colormap type for the X server to use is governed by the defined visual class. There are six visual classes, not all of which can be implemented on all monitors and graphics cards. The following table illustrates this interdependency.

<b>Monitor type</b>	<b>Colormap type</b>	<b>Visual class for color cells which are read/write</b>	<b>Visual class for color cells which are read-only</b>
Monochrome	1 index per RGB value; RGB values identical	GrayScale	StaticGray
Color	1 index per RGB value; RGB values differ	PseudoColor	StaticColor
Color	3-field index for RGB values	DirectColor	TrueColor

Table 4: Interdependency between monitor, colormap and visual class

Even on color monitors it is possible to set the visual class to GrayScale.

GrayScale means that red, green and blue values are converted to shades of gray (grayscale). The gray values in the colormap can be changed dynamically. The actual entry in the colormap has the same value for all three color components.

PseudoColor means that the colormap entry is addressed via a single pixel value. Thus the colormap is evaluated one line at a time, and the combination of red, green and blue values is predefined. There is only one index for each RGB value.

By contrast with PseudoColor, DirectColor means that the values for red, green and blue differ and that there is a 3-field index for them, one field for red, one for blue and one for green.

StaticGray, StaticColor and TrueColor are the static equivalents of GrayScale, PseudoColor

and DirectColor respectively. With the static classes the entries in the colormap cannot be changed.

When the X server is initialized, the root window is generated with a default visual. The same visual is used for the default colormap. The root window bequeaths the properties of the corresponding visual class to its child windows.

The default visual is defined in the X server.

## 3.2 Fonts

A font is a graphical description of a set of characters. Fonts are managed by the X server. When it is started, the X server downloads the font for displaying the mouse pointer, a default font and information on which fonts it can access on the local host. No other fonts are downloaded until specifically requested by a client. If a font is requested which is not available to the X server, it attempts to access the font from the font server specified in its font search path. The X server provides the clients with fonts for outputting text in windows, title bars, menus or in the form of pointer symbols.

A font consists of an array of up to  $2^{16}$  glyphs. A glyph is an image of a character. Font names have by convention a structured format (the X Logical Font Description, or XLFD) providing information about the font's properties, such as font family, character weight, point size, spacing and character set type.

Since X11R5, outline fonts have been supported as well as bitmap fonts. The glyphs of a bitmap font are stored in individual bitmaps with a fixed size. If a font with a different point size is required, separate bit maps must be provided for this size; in other words, each point size necessitates a separate font. The bitmap fonts are converted into a binary format that the X server can read.

With outline fonts, it is not the bit maps of the glyphs which are stored, as is the case with bitmap fonts, but a representation of the contour of each glyph. The X server is then able to interpret this representation so that it can generate all point sizes of the font from one source.

Outline fonts have the advantage over bitmap fonts that they can be scaled quickly and without loss of quality. What is meant by scaling is that different properties can be generated from a font source. The glyphs differ in character height, point size, vertical and horizontal resolution and average character width in other words, in all properties affecting size. Bitmap fonts can also be scaled, but the results are unreliable. The section on "Font naming with the X Logical Font Description " has an example to illustrate how a user can request a scalable font.

### 3.2.1 X server font path

The X server's font path specifies the directories the X server is to search for fonts requested by a client. The font path is scanned from left to right. Within the directories - for the local X server of a workstation they reside in `/usr/lib/X11/fonts` - the fonts are grouped by common properties, for example fonts with the same dots-per-inch (dpi) resolution or fonts of a specific type (such as PostScript Type 1). The misc directory mostly contains fixed-width bitmap fonts (every character is the same width) and the mouse pointer font. With X11R5 and above, you can identify a font server as part of the font path. The font server allows the system administrator to make fonts available to all X servers throughout the network even though they are only stored on one host. The principle of the font server is explained in the section entitled "Font servers ".

Each font is stored in a separate file. The file name extension indicates the format of the font. Bitmap fonts, for example, have the extension `.pcf` (Portable Compiled Font), and the outline fonts from Bitstream Inc. have the extension `.spd`. In each of the directories on the font path, there are two files which act as a database for the X server:

`fonts.dir`

contains a two-column list pairing the names of the files in the directory with the names of the fonts stored in them. If a client or an entry in a resource description file or a command-line argument requests a font by name, the X server reads fonts.dir to find out which font file to use to display the font on the screen.

#### fonts.alias

contains a two-column list pairing font aliases with the matching full font names. If a font is requested by way of its alias, the X server reads fonts.alias to identify the full font name and then reads fonts.dir to find out the name of the file in which the requested font is stored.

The first two entries in the fonts.alias file in /usr/lib/X11/fonts/misc relate to two essential aliases:

**fixed** Whenever the font to be used for text output is left undefined, the X server uses the font assigned to the alias "fixed". You can change this default setting with the X server's -fn option on the client command line or in the appropriate fonts.alias file for all the X server's clients.

#### variable

is the alias for the font some clients use as default font. You should not change the font name allocated to this alias.

### 3.2.2 Font servers

Font servers manage fonts and make them available to other applications, for example to a local X server. This frees the X server from the necessity of storing all fonts on its own host, which can use up a lot of space. A font server typically serves an entire network with fonts, thus reducing redundant data storage.

The "Font Service" desktool supports the system administrator in font server administration tasks. As system administrator you can double-click on it to open a window for font server administration. Using the "File" menu you can identify the configuration file and hence the font server you want to process in the current window. There are three parameters which uniquely identify a font server as a communication partner in the X Window System:

1. The transport protocol used. This is "tcp".
2. The name of the host on which the font server runs. In the case of the font server that you manage with the desktool this is the name of the local host.
3. The number of the port at which the font server expects to receive requests for fonts.

Under the "tcp" transport protocol, the first font server running on a host generally offers its services at port number 7000.

As the transport protocol is "tcp", the desktool shows you only the host name and the port number for the font server you are currently processing. Once you have configured the font server, you can control its activities using the "Font Server" menu.

From the pop-up menu of the "Font Service" desktool you can open a window that makes it easier for you to install individual fonts. For example, when installing a font you can name a font catalog of a font server as target directory. If you then start the associated font server and add it to the font path of the X server you are using, you can also use the new font on your own display.

In the X server's font path, a font server is identified by the three parameters listed above: the transport protocol used, the address of the host running the font server, and the port number.

With the font search path

`/usr/lib/X11/fonts/misc/,tcp/toledo:7000`

for example, the X server finds the fonts on its local host in the directory

`/usr/lib/X11/fonts/misc` and receives all other fonts from the font server, which is run on host "toledo" and accessed via port 7000.

### 3.2.3 Font naming with the X Logical Font Description

X11R3 introduced a font naming convention known as the X Logical Font Description format, or XLFD for short. Since then, font names have featured the following components:

`-adobe-courier-bold-o-normal--10-100-75-75-m-60-iso8859-1`

The meaning of the components in detail:

<code>-adobe</code>	<code>-courier</code>	<code>-bold</code>	<code>-o</code>	<code>-normal</code>	<code>--10</code>	<code>-100</code>	<code>-75</code>	<code>-75</code>	<code>-m</code>	<code>-60</code>	<code>-iso8859-1</code>
											character set
											average character width
											spacing
											vertical dpi resolution
											horizontal dpi resolution
											size in point *10
											character height in pixels
											set width
											slant
											character weight
											font family
											company

Figure 1: Font name components

Not every component need be included when specifying a font name. All font name components can be replaced by asterisks, as in the following example:

`-*-courier-bold-o-***-140-***-***-***`

The asterisk represents a wildcard value for the relevant component. It can also stand for more than one component. The X server uses the first font it finds which corresponds to the specified pattern.

You can recognize a scalable outline font by the fact that the components in the font name which specify the character size have a value of "0", for example:

`-bitstream-charter-bold-r-normal--0-0-0-0-p-0-iso8859-1`

You can request this font from the X server with a character height of 12 pixels by entering the following assignment in a client resource description file:

`*font: -bitstream-***-r-***-12-***-p-***`

From this information, the X server generates the following font, which it then makes available to the client:

`-bitstream-charter-bold-r-normal--12-120-75-75-p-75-iso8859-1`

## 4 Client interaction in SINIX/windows

This chapter describes display, window and desktop management and their interaction with one another and with the X server. It examines client interaction before, during and after a session and in the desktop configuration process.

### 4.1 Client interaction in the X Window System

The X Window System guidelines on interaction between clients are defined in the client development guide entitled "Inter-Client Communications Conventions Manual ICCCM". In keeping with the principles of hardware independence and network transparency, the primary component in client interaction is the X server, which acts as a reliable common interface for the clients associated with it.

#### 4.1.1 Information passing

One example of client interaction is the cut-and-paste mechanism for moving text from one terminal emulator window to another. With terminal emulators, the text you want to cut and paste can be selected a character at a time, a word at a time (with user-definable word boundaries) or a line at a time.

With suitably designed clients you can also move text between clients using the drag-and-drop mechanism. Depending on the particular client you may also be able to exchange other data in the form of objects such as graphics. For more details refer to the manual "SINIX/windows User Environment; XDCL Desktop Configuration Language".

#### 4.1.2 Using window-related information

The mwm window manager is responsible, among other things, for coordinating the sizes and positions of windows and icons for multiple independent clients. Other aspects which require coordination between clients and the window manager include

- the available selection of window frame controls
- input focus assignment and active window designation
- space on the root window
- the position of a window within a window stack
- colors and fonts
- the name on the window frame
- defining the window corner in which window contents are to remain visible after resizing
- requests for buffering clipping regions on the X server

Interaction between clients and the window manager also allows these features to be modified consistently in the course of a session.

To enable X Window System clients to intercommunicate even without a window manager, there are further conventions defining how clients should handle input focus, the mouse pointer and colormaps, how to reserve (grab) X Window System components, and how to query the keymap and modifier map tables on the X server.

## 4.2 Session management

This section provides you with an overview of the progress of a SINIX/windows session, the clients and components involved and the places where you can find further information about the various procedures.

### 4.2.1 Logging in to a session

Client involved:  
xdm

Before you start a session, you must set up a connection for communicating between the X server and xdm.

#### **Note:**

For more information on this topic refer to the section "Display management" and to the description of the X server you use for your display. There you will also find information about which client is responsible for mapping the keyboard, whether the xmodmap client is used for the purpose, and how you yourself can alter how the keyboard is mapped.

Your monitor first either displays a host selection window where you can select a host or it takes you directly to the login window where you can open a session on the host. The xdm display manager displays these windows.

If you are offered a host selection window, you select the host where you want to login. In the factory default configuration you can choose between sessions using the "Session" option menu in the login window. This section describes the procedure for selecting a "user session". In addition, you can select the language in which you want to communicate with the system during the session by selecting the "Language" option button.

#### **Note:**

Information on this subject can be found in the help text for the login window.

Once you have entered your login name and password in the xdm login window, xdm checks your access authorization. If xdm accepts your login name and password, it starts a session for you.

### 4.2.2 Opening a session

Client involved:  
xdm

After you open a session, the xdm client invokes a number of shell scripts, which in turn typically invoke other clients, define environment variables, define a keyboard map and so on.

#### **Note:**

The functions of xdm are described in "SINIX/windows User Environment; Clients

Reference Manual" and in the help screens of the corresponding windows.

xdm also adds to your authorization file an authorization entry defining the valid session key for client access to your display.

**Note:**

This is described in the section "Controlling client access to the X server"..

The system as shipped is configured such that xdm starts each user's first session by invoking the xdesk client to handle desktop and file management, the mwm client to handle window management and the xmessages client to display terminal-related messages associated with the current login name. The desktop manager makes other desktools available and arranges for their graphical presentation to be as defined in the XDCL configuration files.

**Note:**

The section "Managing desktools and files" and section "Window management" discuss the actions associated with desktop and window management.

#### 4.2.3 Changing the session configuration

Clients involved:

xdm, xdesk, mwm

Files used:

session, sw\_session  
xsession, .sw\_profile

The system administrator can specify which clients to invoke and what configuration to invoke them in. He can do this on a system-wide basis or for sessions at groups of displays. To do this, he modifies the xdm shell script /usr/lib/X11/xdm/Xsession. In the factory default configuration, resources are set on a system-wide basis using this shell script and other shell scripts invoked from it. For example, the shell script sw\_session in /usr/lib/X11/siwin invoked here loads the siwin.xrdb resource database from the same directory.

You can define the configuration for your own session using the .xsession file in the HOME directory. If you change the session configuration, you must make sure that all clients in your session know the values you have chosen for the environment variables. Therefore, when you create your own \$HOME/.xsession file, you must define the environment variables in the i \$HOME/.sw\_profile file. The shell script .sw\_profile is executed in the default session configuration. You should only set environment variables there; do not enter calls to alphanumeric applications. The values that you assign, for example the language setting, remain valid throughout the entire session.

**Note:**

You will find information on how to set up a configuration for your own session in chapter "Session configuration using the .xsession file". The section "Interaction during configuration " provides information on the interaction between xdm, xdesk and mwm

when you configure clients which are invoked each time a user logs in.

During your session you may want, for example, to cut and paste text between terminal emulators; or you may drag a file and drop it on a desktool to have it processed in some way. You store the appearance and behavior of the clients you use in configuration files.

**Note:**

The available configuration options are discussed in the chapter "Using resources to customize client appearance and behavior"

During a session you can also make use of the desktools offered by xdesk to customize the appearance and behavior of the user interface according to your requirements. Depending on the tool used and the settings, the adjustments either take immediate effect or they take effect when you start the next application.

#### 4.2.4 Terminating a session

Clients involved:  
xdesk, xdm

In the factory default configuration you can terminate your session by terminating the SINIX/windows desktop. Choose the "Exit..." entry in the "File" pull-down menu to terminate the SINIX/windows desktop.

Session termination is again managed by xdm. At the end of the configuration file for your session you can arrange for typical end-of-session activities to be performed, a daily backup for example. After terminating your session, xdm invokes a shell script in which the system administrator can cause further routine tasks to be performed at the the end of the session. The system administrator can likewise use xdm to ensure that xdm resets the X server when the session terminates.

If you are logged in as the system administrator, you can use the "Shutdown..." entry in the "File" pull-down menu to shut down a system. In the dialog box that appears, you should enter a sufficient delay time for starting shutdown so that you and other local users can terminate any open sessions in the usual way.

## 4.3 Display management

Display management is handled by the client xdm. xdm is the first client with which the user comes into contact, since it opens the login sessions. Other display management tools are described in "Other tools available for display management".

As a display manager, the xdm client performs the following functions:

- It sets up a connection between the X server and the host on which you can log in from your display.
- It displays a login screen with a welcome banner and prompts you for a login name and a password.
- It verifies the login name and password and, if necessary, rejects the login.
- It generates the key for the authorization mechanism.
- It opens the session.
- It configures the session on the basis of configuration files.
- It terminates the session.

Your configuration options for your own sessions are discussed in the chapter "Session configuration using the .xsession file". The authorization mechanism is described in the section "Controlling client access to the X server ". The xdm client is described in full in the "Clients" chapter in "SINIX/windows User Environment; Clients Reference Manual".

### 4.3.1 Connection setup between X server and xdm service

The display manager is capable of managing multiple displays both on the local host and on remote hosts. Each display is controlled by an X server. Before xdm can manage a display, a connection must be set up between xdm and the X server. The initiative for setting up the connection can be taken either by the X server or by xdm. X servers for X terminals generally start connection setup on their own initiative. For connections initiated by the X server, the X server and the xdm service use the XDMCP protocol. XDMCP is supported by X servers and the xdm written with X Window System version X11R4 and above. X servers not supporting this protocol must be managed directly by xdm via an entry in its Xservers file. The local X server is usually started by xdm via an entry in its Xservers file.

X servers with access to the XDCMCP protocol can call up xdm on their own initiative and request it to manage the display they control. xdm reads its Xaccess file to find out which X servers are allowed to make use of a host's xdm service.

X servers distinguish between three types of XDCMP request: direct, broadcast and indirect requests.

- Direct requests: the X server requests the xdm service of a specific host on the network. If the X server is listed in the Xaccess file on that host, or is not explicitly denied access there, the display will be managed directly by the host's xdm and will be sent a login window.
- Broadcast requests: the X server sends an XDMCP request to all the hosts on the network. All active xdm services check the server's authorization. The first xdm service to respond takes over display management.
- Indirect requests: the X server sends an indirect XDMCP request to a specific host. That host's xdm checks the entries in the host's Xaccess file and on that basis decides whether to accept the request and which hosts to forward the indirect request to. The first host to respond takes over display management.  
If the entry for the display contains the keyword CHOOSER, the user at the display will be given the chance to select a host to log in on. xdm does this by starting the Chooser

application. This passes on the request to all hosts listed after the keyword CHOOSER in the Xaccess file, or to all hosts in the network if the keyword CHOOSER is followed by the entry BROADCAST instead of the host names. The addressed hosts reply with a positive or a negative acknowledgment, depending on the entries in their Xaccess files.

The user is sent a host selection window containing the names of the hosts which sent a positive acknowledgment. The user can select a host from the icons in this window. xdm's Xservers file contains the X servers to which xdm is to set up the connection itself. No display may be entered in this file if its X server contacts xdm directly via XDMCP. X servers of version X11R3, on the other hand, must be made known to xdm via this file, since they do not support XDMCP.

xdm can start the X server of the local host via a command in the Xservers file. At regular intervals xdm attempts to set up a connection to other X servers entered in the file.

#### **4.3.2 Other tools available for display management**

- The xhost client: this is used to grant or deny access to the X server for clients on remote hosts (on a host-specific basis).
- The xauth command: this is used to transfer the user-specific authorization information from the authorization file of a user to remote hosts. This means that access to the X server can be granted or denied for clients on remote hosts.
- The desktool "Lock & save screen": this is used to lock access to a screen.

## 4.4 Window management

The window frames of all clients allow you to perform various window control actions, such as enlarging, reducing, moving or iconizing. These actions are handled by the Motif window manager (mwm), and they are described in the manual "SINIX/windows User Environment; Introduction to Handling and Configuration". The client is responsible for updating the window contents, while mwm controls the window frame and hence all the actions it can be used to initiate. The window manager provides a pop-up menu on the root window and the window menu accessed via the window menu button on the window frame. The chapter "Customizing mwm window manager menus " contains more detailed information on configuring mwm menus.

When you open a session, mwm is automatically started by the xdm client in the configuration as shipped with the product.

As mwm plays a central part in passing on keyboard and mouse input, the table below lists all the mwm resources which have an impact on input focus assignment. For a full description of the resources and their values you should refer to "SINIX/windows User Environment; Clients Reference Manual". The chapter "Using resources to customize client appearance and behavior" in this manual shows you how to change the behavior of the mwm client using the resources listed below. Alternatively, for many of the settings you can also use the window manager desktool provided with the xdesk client.

Functionality	Topic	Resource
Defining input focus policy: explicit = click to assign, implicit = position of pointer	General policy	keyboardFocusPolicy Class KeyboardFocusPolicy
Defining whether window stacking order changes when input focus is reassigned	Stacking order	focusAutoRaise Class FocusAutoRaise
Delay in milliseconds before mwm raises window assigned input focus to top of window stack	AutoRaise delay	autoRaiseDelay Class AutoRaiseDelay
Defining whether explicit input focus assignment policy is to apply to all windows	All windows	enforceKeyFocus Class EnforceKeyFocus
Defining whether window becoming visible on screen is assigned keyboard input focus	Opening windows	startupKeyFocus Class StartupKeyFocus
Defining whether mwm automatically sets input focus when window is normalized from icon size	Normalize from icon	deiconifyKeyFocus Class DeiconifyKeyFocus
Defining whether window normalized and raised to top of window stack by mwm f.normalize_and_raise	Normalize and raise	raiseKeyFocus Class RaiseKeyFocus

function is assigned input focus		
Defining input focus assignment policy when active window is closed/minimized	Close/minimize active window	<b>autoKeyFocus</b> <b>Class AutoKeyFocus</b>

**Table 5: Customising the window manager**

## 4.5 Managing desktools and files

The desktop is the graphical workspace on which you can place the objects you work with, such as desktools and files. The objects are represented by graphical symbols. Desktools are managed by the desktop manager, files and directories (folders) by the file manager. The desktop manager and the file manager are integrated under the xdesk client, also known as the SINIX/windows desktop. Using the xdfm client you can start the file manager without the desktop manager. During your session the desktop manager and the file manager let you access desktools and use them to process files and directories.

As a desktop manager, the xdesk client performs the following functions:

- It generates and manages the desktop, either in a separate window or on the root window.
- It manages the desktools. The graphical image (icon) representing a desktool on the desktop gives you access to an application, a shell script or internal methods of the desktop manager.
- It provides you with a consistent working environment in which you can perform common operations by graphical interaction.
- It supports the drag-and-drop mechanism for desktools.
- It allows you to configure desktools interactively. For example, you can specify which desktools should be visible and define their graphical presentation and their arrangement on the desktop.
- It enables you to start a session with your desktop configured as it was at the end of the last session. It displays the desktools present then in the same positions within a separate window or on the root window, provided that the window manager will allow it to do so. However, it does not restore the former status of the desktools: for example, the status of current applications is not saved.
- It allows you to create and configure desktools using the XDCL configuration language.

By graphical interaction you can find out useful information about your system and the current X server from the "Information" desktool. You can open "Mail" and, depending on the mode, read a file using the drag-and-drop mechanism or mark files for mailing; you can drag and drop files on the appropriate desktool to load them into your editor, copy files from one directory window to another - even moving them from one host to another - and do all kinds of other things. Interactive configuration and the operation and functionality of the desktools are described in the help texts for each component of the user interface.

The XDCL configuration language can be used to define other desktools for your login, for a group of displays or globally for the entire system. Configuration with XDCL is the subject of the manual "SINIX/windows User Environment; XDCL Desktop Configuration Language".

The file manager is contained in the desktools catalog and together with the xdesk client is made available by the xdesk client. However, it is also possible to start the file manager without the xdesk desktop manager, using the xdfm client. You can start one instance of the file manager per user and display on each host. The file manager performs the following functions:

- It manages one or more directory windows.
- It allows you to manage your files and directories on the local host. For example, you can create, copy and delete files and directories, change file and directory permissions, or find out about file sizes.
- It allows you to interact with file managers on remote hosts so that you can also manage your files and directories on other hosts in the network.
- It allows you to define file types on the basis of the file name extension or of entries from

the `/etc/magic` file.

- It allows you to configure views on file and directories in directory windows. For example, you can define the appearance of file types differently for different directory windows, even specifying whether files of a particular file type should be visible in a directory window.
- It associates a file type with an application. Then in response to a double click on a file of the appropriate type it will start the corresponding application and pass it the selected file to operate on. Bitmap files, for example, are associated with a pixmap editor.
- It makes files and directories available to you in the form of objects on the desktop. Where appropriate you can then drag and drop an object to have the file or directory it represents processed by an application. Thus if you drop a file on the "Trash" desktool, it will be consigned to the trashcan.
- It allows you to configure associations between files and applications, either interactively or using the XDCL configuration language.

For details on how to operate the desktop and file managers, see the appropriate help texts.

#### 4.5.1 Drag and drop for desktop objects

Desktop managers and file managers are typical examples of the drag-and-drop mechanism for graphical interaction with objects.

The file manager supports graphical, interactive management of files and directories. It treats files and directories as objects which it represents symbolically on your desktop, where you can manipulate them using drag and drop. For example, you can move a file to a directory by dragging it and dropping it onto a directory icon. You can also copy files from one file manager window to another one on a different host.

Another example of an "object manager" like the file manager is the desktop manager, which makes the desktools available in the form of objects. As a desktop manager it knows which actions (methods) are associated with dropping objects on desktools; so provided a drag-and-drop method associating the selected object with the selected desktools has been defined, it will start the appropriate action when the object is dropped on the desktool. For the desktop manager, drag and drop means that one object is dragged on to another object and dropped there, causing the first object to be processed by the second.

## 4.6 Interaction during configuration

This section examines the tasks performed by the clients and the X server in the course of in-session configuration. The tasks are tabulated below in the order in which they affect your sessions. The associated key terms (under "Topics") act as an index to the documentation listed under "Refer to...", enabling you to locate the information about a task that you will need if you want to customize the configuration. References to clients relate to the corresponding client description in "SINIX/windows User Environment; Clients Reference Manual". References to your X server's display relate to the documentation for your X server.

The following overview does not cover graphical interactive desktop configuration options, as these are dealt with in the help texts for the relevant components of the product.

<b>Task</b>	<b>Topics</b>	<b>Refer to...</b>
Connection setup between X server and xdm	for X11R3: xdm Xservers file, as of X11R4: XDMCP as of X11R5: xdm's Xaccess file	xdm client, your display's X server
Appearance and behavior of host selection window	Xaccess and Xresources file of xdm	xdm client
Icons in host selection window	host manager, host, icon	xdm client, help texts for xdm host manager
Configuring keymaps, pointer maps	modifier keys, xmodmap, xdm Xinit file	your X server's display, xmodmap client
Icons in the xdm login window	user manager, user, icon	xdm client, help texts of xdm user manager
Appearance and behavior of xdm login window	xdm Xresources file	xdm client
Keyboard-activated functions in xdm login window	xdm Xresources file, translations resource	xdm client
Preparations directly after output of xdm login window	Xsetup file of xdm	xdm client

Access to X server from other hosts	X server access control list, DISPLAY environment variable, host-specific authorization	your X server's display xhost client, section "Controlling client access to the X server "
Client access to X server	user-specific authorization	xdm client, your X server's display, section "Controlling client access to the X server "
Preparations straight after system login, e.g. mounting file systems	xdm Xstartup file	xdm client
Starting first client after system login, defining environment variables, system-wide and display and user-specific session configuration	xdm Xsession file, .xsession	xdm client, section "Session management", chapter "Session configuration using the .xsession file"), desktool help texts.
If you work with the configuration as shipped, you can define your environment variables in your \$HOME/.sw_pro file file	.sw_profile, environment variables	section "Session management"
Selecting an international environment on a system-wide or user-specific basis	LANG in xdm-config file of xdm, "International" desktool	section "Modifying the country-specific user environment"
Window manager startup	root menu, window menu, input focus, active window	mwm client, chapter "Customizing mwm window

		manager menus "
Desktop startup, system-wide or user-specific desktop configuration	desktop manager, file manager, user manager, mail manager, host manager, XDCL Desktop Configuration Language	xdesk, xdfm, xdhm, xdm, xdum clients, section "Managing desktools and files" and manual "SINIX/windows User Environment; Desktop Configuration Language XDCL".
Starting other clients	desktop, root menu, .xsession, resource, XENVIRONMENT environment variable, .Xdefaults file	client descriptions, description of resource evaluation hierarchy under "File hierarchy" in "SINIX/windows User Environment; Clients Reference Manual", chapter "Using resources to customize client appearance and behavior".
Defining client appearance and behavior	resource, XENVIRONMENT environment variable, siwin.xrdb file in /usr/lib/X11/siwin, .Xdefaults file, .Xresources file	client descriptions, especially those for xrdb and xdm, description of resource evaluation hierarchy under "File hierarchy" in "SINIX/windows

		User Environment; Clients Reference Manual", chapter "Using resources to customize client appearance and behavior".
Session termination	Closing the desktop	xdesk, xdm clients
Resetting the X server	xdm Xreset file	xdm client, your display's X server

Table 6: Interaction during configuration

## 5 Session configuration using the .xsession file

If you want a configuration of your choice to be operative from the start of a session, you must enter all the instructions that have to be carried out at session startup in a file named `.xsession` residing in your home directory. The commands in this file are executed whenever you start a session. The `.xsession` file contains a shell script.

In this file you can

- set environment variables,
- invoke clients,
- define the appearance of the root window.

The `.xsession` file in your home directory is called from `xdm`'s `Xsession` file and it overrides any subsequent entries there which specify the configuration of user sessions.

Consequently if you create a `.xsession` file, it is up to you to ensure that all the essential instructions for your session are in it. Thus if you create an empty file, or if the instructions in the file cannot be carried out, no session will be started for you.

### 5.1 Sample .xsession configuration file

You want the following configuration to take effect when you start a session:

- The `PATH` environment variable is to be set to include the client invocation path.
- The `mwm` window manager is to be started.
- The `xterm` terminal emulator is to be started.
- Another terminal emulator of type `xpct` is to be started. The terminal emulator window is to come up showing the `vi` editor (or an editor of your choice, of course) with the `.xsession` file loaded. When you quit the editor, the window is to be closed as well.

Create a `.xsession` file consisting of the following lines:

```
PATH=$PATH:/usr/bin/X11:.
export PATH
mwm &
xpct -e vi .xsession &
xterm -geometry 80x30+10+10 -title "Hi Everyone" -ls \
-fg black -bg white -cr white -sb -sl 30
```

Once you have put in all the instructions you require, save the file. The changes come into effect when you terminate your session and start another one. The user interface will then be arranged in accordance with the instructions in your `.xsession` file.

If you are not quite sure whether all the entries in the file are correct, you may proceed as follows:

- terminate all clients apart from a terminal emulator
- run the `.xsession` file by entering the command `. $HOME/.xsession`

If this fails, you will be left in the terminal emulator, where you can review and modify the entries in `.xsession`.

If your session starts up as intended, you can carry on modifying the `.xsession` file until you have the configuration exactly as you want it. If you do not want to make any more changes, you should quit the editor. That will also close the terminal emulator in which you started it.

## 5.2 Points to note

To avoid problems when working with the `.xsession` file, you should note the following points:

- The user-specific session management configuration in the `$HOME/.xsession` file overrides any system-wide session management specifications. Thus if you create a `$HOME/.xsession` file, you may need to make it start the mwm window manager and the xdesk desktop and file manager, for example.
- Set the environment variables (such as `PATH`) before starting clients, as otherwise the clients will not evaluate them.
- In the `.xsession` file, run clients in the background by appending "&" to the command line. This does not include the last client in `.xsession` that outputs a window. Also excluded are clients which configure attributes for the X server, e.g. the `xrdb` client. If you do not run a client in the background, the next client will not be started until the previous one has been terminated.
- The session is closed when the last client not run in the background is terminated. Any clients started after it as background processes are instantly terminated as well. You need to allow for this when defining the order of the entries in the file. It would be most unhelpful for the last entry to be, say, a client which shuts itself down once it has performed its function. The session would then die with it. It is generally best for the last entry to be the terminal emulator you intend to use for entering commands.
- If a login shell is started, other startup scripts (such as `.profile`, `.cshrc` or `.login`) may be executed in addition to `.xsession`, depending on how the operating system is configured. If these files contain conflicting instructions, the instruction that will apply to the shell that is invoked is the one in the script executed last.

### Example:

Entry in the `.xsession` file:

```
PATH=$PATH:/usr/bin/X11
    # Sets the PATH variable ...
xterm -ls
    # Starts xterm with a login shell
```

Entry in the `.profile` file:

```
PATH=/bin:/usr/bin
    # Sets the PATH variable ...In this case the value assigned to the PATH
    variable in .profile overwrites the value assigned to it in .xsession.
```

- If you always work with the graphical user interface under your login name and want to start your own choice of clients, you should define all such settings in the `.xsession` file only. When doing this, remember that, for example, `xdm` assigns a value to some environment variables - e.g. the `PATH` environment variable - at the very beginning of the session.
- If you create your own `$HOME/.xsession` file, you yourself may have to ensure that all clients in your session know the values you desire. This applies to environment variables as well as to resource value assignments. In the configuration as shipped with the product, resources throughout the system are set via `xdm`'s `Xsession` file and shell scripts called from there. The `sw_session` shell script in `/usr/lib/X11/siwin` called

there, for example, loads the resource database `siwin.xrdb` from the same directory.

## 6 Using resources to customize client appearance and behavior

Resources are the primary tools for customizing the user interface. Resources let you modify the appearance and behavior of clients.

The resource mechanism was designed to provide developers, users and system administrators with a simple and convenient tool for long-term user-friendly configuration of the interface.

You can influence the appearance and behavior of a client by using command-line options when calling the client; but it is far easier to use resources. The resource mechanism allows specifications to be defined for several clients in common and automatically put into effect whenever the client is called.

Resources are also used because many clients have a range of features which you can customize to suit your own preferences. The client invocation syntax would grow totally confusing and the command lines infinitely long if you were to define everything on the command line, repeating the whole procedure every time you called the client. You can define a long-term value for a resource in a file, and this value will then be applied automatically whenever the client is called. Of course, you can also change these predefined values when you call the client: value assignments on the command line override value assignments in files.

The following is a list of some client characteristics that you can define using resources:

- colors and patterns for window foreground and background, window frame, menus, icons
- fonts for text in windows, window frames, menus, icons
- 3-D shadows
- decoration and layout of windows, icons, menus
- input focus policy for input devices
- size and position of windows, icons
- names of configuration files

Using resources you can, for example, tell mwm to use a particular font for displaying the window title, to display the title in dark blue letters, and so on.

Resources are grouped by function in resource classes. Each resource belongs to a resource class. Resource classes allow a number of resources to be assigned a common value. A resource class name begins with an uppercase letter, a resource name with a lowercase letter.

Thus there is a Foreground resource class, comprising a whole range of resources for defining foreground colors. The cursorColor resource governs the color of the text cursor, the iconImageForeground resource governs the foreground color for icon images, and so forth. You can assign the value "red" to all resources of resource class Foreground by simply assigning the value "red" to resource class Foreground. However, you can also assign the individual resources different values.

Thus the assignments

```
*Foreground: red
*cursorColor: green
*iconImageForeground: yellow
```

specify that all clients which support a resource from resource class Foreground should use red for their foreground color, the exceptions being the colors for the cursor and the icon image, which are assigned different values.

Clients are shipped with defaults for all their resources. These defaults can be changed in various ways, either directly by the user, or by the system administrator or a client programmer.

At the request of a client the X server will itself also keep the values of of the resources named by the client in a resource database. On being started and before appearing on the screen, a client asks the X server whether there are any resource definitions for it in configuration files.

In this chapter we illustrate various ways in which you as user can define resources for clients.

In particular you will find:

- an introduction to resource syntax
- how to assign a value to a resource
- an introduction to the hierarchy of value assignments

## 6.1 Resources

A resource definition for a chosen component may consist of up to four elements:

- the name of a client to which the resource definition is to apply.  
If no client name is specified, the definition applies to all clients which support the defined resource.
- the name or names of a subcomponent of the specified client.  
This name identifies the client subcomponent to which the resource definition is to apply. A subcomponent may be a specific window or a menu or some other element which the client controls. If no subcomponent is specified, the definition applies to all client subcomponents which support the defined resource.
- the resource name.  
This identifies the client attribute to which the resource definition is to apply. This element is mandatory.
- the value to be assigned to the resource. This element is also mandatory.

Thus a resource definition may look like this:

```
mwm*icon*foreground: yellow
```

This resource assignment specifies that the mwm client should use yellow as the foreground color for its subcomponent "icon". Thus among other things the names on icons managed by mwm will appear in the foreground color.

Depending on how precisely you define your resource, you can influence different numbers of components of the interface.

Below we exemplify this using a resource which defines the font to be used for text in windows and on window frames. The name of this resource is fontList. It belongs to the FontList resource class.

We want to tell mwm to use a font named *fontname* to display client titles in window frames. One way to do this would be:

```
*fontList: fontname
```

This would have the desired effect, but it would also cause font *fontname* to be used by all clients for all text in all windows and all window frames. This would apply to all clients which support the fontList resource, such as mwm and x97801-480.

If that is not what you want, you need to make your resource definition more precise.

```
mwm*fontList: fontname
```

Now the definition applies to the mwm client only. This value assignment specifies that font *fontname* is to be used by the mwm client for all text in mwm window frames and menus. You can make your definition even more restrictive.

```
mwm*client*fontList: fontname
```

This value assignment specifies that mwm is to use the font for all text in client window frames, i.e. for the title and the window menu but not for application icons or the root menu. However, we wanted our definition to apply to the title only.

```
mwm*client*title*FontList: fontname
```

This value assignment finally specifies that font *fontname* is to be used by the mwm client for all titles in client window frames. Any other text displayed by mwm is not affected.

This example shows that working with resources offers many opportunities for explicitly influencing client behavior. There will be more examples to demonstrate this later.

First, though, we describe exactly how to assign a value to a resource.

## 6.2 Assigning values to resources

To use resources to customize the user interface to your own requirements, you have to assign values to the resources.

If there is no entry defining a resource in a configuration file and no value is assigned to it in the client call, the client uses the default.

The approach you adopt to assign values to resources depends on the resources you want to customize and on whether you want to redefine them permanently or just for a limited period.

Resource values can be assigned:

- in options on the client command line.  
If you simply want to make a temporary change to the value of a resource, assign the value on the client command line. This is also a useful approach if you invoke clients in shell scripts and customize your interface in that way.
- in entries in configuration files.  
If you want to make a long-term change to a resource value, assign the value in a resource configuration file.

This section describes:

- how to find the right resource for the attribute you want to change
- assigning values to resources in options
- assigning values to resources in files

### 6.2.1 Resource selection

Before you can change any attribute of a client in your user interface, you need to find out which resource controls the chosen attribute. To do this you will require the manual "SINIX/windows User Environment; Clients Reference Manual", which gives details of all the clients and resources available to you.

- First decide on the client for which you want to modify a resource.  
Remember that mwm is responsible for the appearance of window frames. In other words, the appearance of window frames is controlled by mwm resources. The same applies to the root menu and the icons.
- Consult the client description to find out which resources are available, select the appropriate resource and establish which resource class it belongs to.
- Then, in the chapter headed "Resources", first locate the resource class (arranged alphabetically) and then the resource itself (arranged alphabetically within the class).  
There you will find the syntax and the range of acceptable values.

In the following example, the icons are to be placed in an icon box. Icons are controlled by the mwm window manager.

1. Consult the mwm client description to find out which resource controls the icon box. The resource name is `useIconBox`, the class is `UseIconBox`.
2. In the chapter headed "Resources", locate the description of the `useIconBox` resource in the `UseIconBox` class. There you can establish what value you need to assign to the resource to make mwm use an icon box. The appropriate value is "True".

You now have all the information you need to define a resource. The assignment you require is:

```
mwm*useIconBox: True
```

Next you must decide where to use this assignment. Your value assignment can either be included in the command line when you invoke mwm or stored in a configuration file.

### 6.2.2 Assigning values to resources in options

You will find the description of the options supported by a particular client in the description of that client in the reference manual for your user interface.

The clients support value assignments to resources in

- the `-xrm` option,
- options which assign a value to a specific resource.

Using the `-xrm` option you can assign a value to any resource that a client supports.

Format: `-xrm "resourceDefinition:value"`

If you want to try out the two examples below, you must first terminate the current instance of `mwm`. One way to do this is to use the operating system's kill command.

Then the correct call for our "icon box" example is:

```
mwm -xrm "*useIconBox: True" &
```

The option must be entered separately for each resource that is to be set. Thus if you also want to give the icon box the title "The Icon Corral" (resource: `iconBoxTitle`; class: `IconBoxTitle`), the call to `mwm` must look like this:

```
mwm -xrm "*useIconBox: True" -xrm "*iconBoxTitle:
The Icon Corral" &
```

To simplify resource definition syntax, some clients support options which set resource values directly.

Say you want to specify that text in `xterm` emulator windows is to be displayed in the color "red". This attribute is controlled by the foreground resource of resource class `Foreground`.

The `xterm` client supports the `-fg` option for this purpose, so you can enter:

```
xterm -fg red &
```

You could achieve the same result with:

```
xterm -xrm "*foreground: red" &
```

You can also combine the two forms of resource definition on one command line. The command line

```
xterm -fg red -xrm "*background: blue" &
```

sets the foreground color to "red" and the background color to "blue".

### 6.2.3 Assigning values to resources in files

There are a series of files in which resources can be defined. The following approach is recommended for users wishing to customize the resources for elements of the user interface:

- Create a file, e.g. `own-config`, in your home directory and store your resource value specifications there.
- Assign the name of this file to the `XENVIRONMENT` environment variable, e.g.

```
XENVIRONMENT=${HOME}/own-config; export XENVIRONMENT
```

In this file you can assign resources your preferred values for use by all clients for the duration of one of your sessions.

If you work with various hosts in the network, you may have to specify values for different displays with differing hardware characteristics (e.g. color capabilities). The convenient way to manage this is to

- create a separate file *X-displayhostname* for each host supporting a display on which you work,
- each time you change to another host, assign the `XENVIRONMENT` environment variable the appropriate value of *X-displayhostname*.

The default value of `$XENVIRONMENT` is `$HOME/.Xdefaults-hostname`.

In the following example of resource definitions in `$XENVIRONMENT` we have resorted to a number of examples introduced previously and added a few new ones.

You want to make the following changes to the default behavior of various clients for your sessions:

- Behavior of mwm
  - An icon box is to be used (useIconBox resource of class UseIconBox).
  - The title on the icon box is to be "The Icon Corral" (iconBoxTitle resource of class IconBoxTitle).
  - The xpct icon is to be generated from the bitmap file "xpct-icon", which resides in `$HOME/bitmaps` (iconImage resource of class IconImage).
  - The window on which the mouse pointer is positioned is always to be made the active window and assigned the input focus. By default a client window only gets the input focus when the mouse pointer is in the window and the <BSelect> mouse button is clicked there (keyboardFocusPolicy resource of class KeyboardFocusPolicy).
  - In addition, the active window in the window stack is to be raised to the top whenever the mouse pointer dwells on it for longer than 20 milliseconds (resources focusAutoRaise of class FocusAutoRaise and autoRaiseDelay of class AutoRaiseDelay).
- Behavior of xterm terminal emulator
  - The color in which text is to be displayed is "white" (foreground resource of class Foreground).
  - The background color is to be "black" (background resource of class Background).
  - The title is to be: This is an xterm window (title resource of class Title).
- Behavior of x97801-480 terminal emulator
  - The shell invoked is to be a login shell (loginShell resource of class LoginShell).

Enter the following in your `$HOME/own-config` file:

```
mwm*useIconBox: True
mwm*iconBoxTitle: The Icon Corral
mwm*xpct*iconImage: ~/bitmaps/xpct-icon
mwm*keyboardFocusPolicy: pointer
mwm*focusAutoRaise: True
mwm*autoRaiseDelay: 20
xterm*foreground: white
xterm*background: black
xterm*title: This is an xterm window
x97801-480*loginShell: True
```

To make sure that the entries in your "own-config" file are evaluated, you must ensure that

- the "own-config" file is read
- the mwm, xterm and x97801-480 clients are started; and when they are started, no other specifications for the same resources must be defined in options.

In addition, you should start the xpct client to check the use of the xpct icon.

Thus you should change the entries in your `$HOME/.xsession` file:

```
PATH=$PATH:/usr/bin/X11:.  
XENVIRONMENT=$HOME/own-config  
export PATH XENVIRONMENT  
mwm &  
xpct &  
x97801-480 &  
xterm
```

**Now terminate all active clients (including mwm) except for one terminal emulator and run the .xsession file. The clients entered in .xsession will be started and will use the resource definitions entered in your own-config file.**

### 6.3 Hierarchy of resource value assignments

You have learnt that there are various ways of assigning values to resources, for example in a file, directly with an option, or with the `-xrm` option.

If one resource is assigned values in a number of places, the values are evaluated in a defined order of precedence. The precedence of a value assignment is derived from a two-stage hierarchy:

- a hierarchy among configuration files containing your assignments of values to resources,
- a hierarchy among entries within these configuration files.

The hierarchy is governed by the following rules:

1. Value assignments to resources in options you specify when calling a client override all other value assignments to those same resources. The other value assignments may be, for example, entries in configuration files.

This applies provided that the level of precision in the resource definition in the option is at least equal to that used in the file.

Thus the resource definition

```
mwm -xrm "mwm*xterm*foreground: red"
```

in the client call overrides the following resource definition in `$XENVIRONMENT`:

```
mwm*xterm*foreground: blue
```

But:

In relation to `xterm`, the resource definition

```
mwm -xrm "**foreground: red"
```

does **not** override the following entry in `$XENVIRONMENT`:

```
mwm*xterm*foreground: blue
```

To be more accurate, the `-xrm` option sets the foreground color in the frames of all windows to "red" (`mwm -xrm "**foreground:red"`), except in the frames of `xterm` windows. The more precise specification in the `$XENVIRONMENT` file (`mwm*xterm*foreground: blue`) sets the color for `xterm` windows to "blue".

This rule also applies to any entries which may have been made in other files, such as configuration files set up by the system administrator, over which you as user have no control.

2. The greater the precision of a resource definition, the higher its precedence.

In relation to the `xterm` client, the entry

```
mwm*xterm*background: white
```

overrides the entry

```
mwm*background: black
```

To be more accurate, the background color in window frames is "black" (`mwm*background: black`), except for the window frames of `xterm` windows (`mwm*xterm*background: white`).

This rule also applies regardless of where the resource is defined, whether in a file or in an option of any type.

## 7 Customizing mwm window manager menus

This chapter describes how to use mwm configuration files to customize mwm menus such as the root menu or the window menu and how to define new menus. To access these menus you need mwm key and button bindings, and this chapter also describes how these can be customized.

The syntax for menus and for key and button bindings and the mwm functions that you use are all described in the section dealing with the mwm client in "SINIX/windows User Environment; Clients Reference Manual".

### Menus

Menus provide you with an easy way of executing functions that you use repeatedly. A menu is a list of functions. With mwm, the functions you can use are the mwm functions. Each element in the list has a name, i.e. the menu item. The list also has a name, the menu name.

The list of functions is displayed in a window, also referred to as menu, when a predefined key or button sequence is entered. The window determines its own size, depending on the font used and the number and length of the menu items. You can then select an item from the menu using another predefined key or button sequence.

### Key bindings

Key bindings provide you with an easy way of executing frequently used functions. Only one set of key bindings can be in effect at any one time.

Key bindings "bind" functions to certain key sequences, i.e. you bind the action "press a key sequence" to an mwm function. A key sequence can be a single key or a combination of keys.

A set of key bindings is a number of key bindings grouped together under one name. This allows you to enable all the key bindings that have been grouped together under that name simultaneously using the keyBindings resource instance (class: KeyBindings). If it is clear from the context whether a single key binding or a set of key bindings is meant, the term "key binding" is used in both cases.

### Button bindings

Button bindings provide you with an easy way of executing frequently used functions with the aid of the mouse buttons. Only one set of button bindings can be in effect at any one time.

Button bindings "bind" a certain action performed using a button sequence to a function. A set of button bindings is a number of button bindings grouped together under one name. This allows you to enable all the button bindings that have been grouped together under that name simultaneously using the buttonBindings resource instance (class: ButtonBindings). If it is clear from the context whether a single button binding or a set of button bindings is meant, the term "button binding" is used in both cases.

A button sequence can involve either a single button or a button used in conjunction with one or more modifiers.

mwm recognizes the following button sequence actions:

Action	Description
--------	-------------

n	
down	Press button down
up	Release button after having pressed it down
Click	Press button and release it
Click2	Press and release button twice. The amount of time that may elapse between the clicks is specified in the doubleClickTime resource instance (class: DoubleClickTime).

Table 7: Button actions

You can, for example, bind the action "press down mouse button <BSelect>" to an mwm function.

## 7.1 How to define a menu

The following is a list of the steps involved in defining a menu.

- Put together a list of the commands that you use frequently.  
Decide on the following for each command:
  - a suitable short name to be displayed in the menu window as the menu item. The name can contain blanks.  
You can also use a bitmap instead of a name.  
The menu item should be easy to remember and should be of a similar nature to the other selections on the menu.
  - the mwm function and the appropriate arguments needed to execute the command.  
This is the function that is executed when you select the menu item.  
Under the heading "mwm functions" in the mwm client description you will find a list of the mwm functions available. The mwm functions are described in alphabetical order.
- Decide on a suitable short name for your new menu; this is the menu name.  
The name should be easy to remember and should be of a similar nature to the other menus forming part of your user interface.

Up to now you have decided on all the menu elements needed to define an mwm menu. You can skip the next two steps if you are not interested in providing a quick way of selecting menu items.

- Decide which menu items you want to be able to select by entering a single letter on the keyboard when the menu is open (visible).  
Decide on a mnemonic, i.e. letter of the alphabet, for each menu item. This mnemonic must uniquely identify the menu item and be one of the letters making up the name. Please note that if the mnemonic is an uppercase letter, you must enter the mnemonic in the menu definition as an uppercase letter; the menu item must, however, be selected using a lowercase letter. Uppercase letters are not accepted as input when making a selection from a menu.
- Decide which menu items you want to be able to select quickly using a key sequence even if the menu is closed, i.e. not visible. This key sequence is also referred to as the accelerator or accelerator string of the corresponding menu item.  
For each of these menu items, decide on an accelerator string which uniquely identifies the appropriate menu item in the context in which the menu could be displayed.

Enter the key binding in the file, using the syntax described under "accelerator" under the heading "Menu syntax" in the mwm client description.

You must perform the following steps in order to define the menu for which you have already determined the menu items.

- The next thing to do is enter your menu definition in the file whose name you passed to mwm as the value for the configFile resource instance (class: ConfigFile).

Enter the menu definition in the file, using the syntax described under the heading "Menu syntax" in the mwm client description.

The next steps involve defining the key or button sequences used to display the menu.

In other words, you need to define a key binding (section "How to define key bindings ") or a button binding (section "How to define button bindings "). Below you will find a few additional points to remember when defining a menu.

- Select a mouse button or key sequence which is to be used to post the menu.
- Select the action to be performed with a mouse button in order to post the menu. As keys can only be pressed, you cannot select a key action.
- Bind the key or button (with the selected action) to the mwm function `f.menu menu_name` or `f.post_wmenu`.

You have now defined a new menu.

## 7.2 How to define key bindings

The following describes the individual steps involved in defining key bindings.

- Compile a list of the commands you use frequently.
- Decide on a name for this list. This is then the name of the key binding.
- For each command decide on an appropriate short accelerator string which can be used to execute the command. The accelerator should be easy to remember and compatible with other accelerators. Each accelerator must be unique within the context in which it is used.

Key sequence syntax is described under "accelerator" under the heading "Key binding syntax" in the mwm client description.

- Work out which mwm function (including the appropriate arguments) is required to execute each command. This is the function that is to be executed when you enter the accelerator.

You will find a description of the available mwm functions under the heading "mwm functions" in the mwm client description. The functions are described in alphabetical order.

- Decide on the context in which the key binding is to apply. The context indicates which components have the keyboard input focus. The following contexts may be specified :

context	component
root	root window
icon	icon
app	client window including window border and titlebar
window	client window including window border and titlebar
frame	client window including window border and titlebar
title	client window including window border and titlebar
border	client window including window border and titlebar

Table 8: Contexts for key bindings

- Next you enter your key binding in the file whose name you passed to mwm as the value for the configFile resource instance (class: ConfigFile). Enter the key binding in the file, using the syntax described under the heading "Key binding syntax" in the mwm client description.
- Specify the name of the key binding as the value for the keyBindings resource instance (class: KeyBindings) as described in the chapter "Using resources to customize client appearance and behavior".

You have now defined a new key binding.

### 7.3 How to define button bindings

The following describes the individual steps involved in defining button bindings.

- Make a list of commands you use frequently.
- Decide on a name for this list. This is then the name of the button binding.
- For each command decide on an appropriate short button sequence that can be used to execute the command. The button sequence should be easy to remember and compatible with other button sequences. Each button sequence must be unique within the context in which it is used.

Button binding syntax is described under "button sequence" under the heading "Button binding syntax" in the mwm client description.

- Work out which mwm function (including the appropriate arguments) is required to execute each command. This is the function that is to be executed when you enter the button sequence.

You will find a description of the available mwm functions under the heading "mwm functions" in the mwm client description. The functions are listed in alphabetical order.

- Decide on the context in which the button binding is to apply. The context indicates which component the pointer must be positioned on for the button sequence to apply. The following contexts may be specified:

context	components
root	root window
icon	icon
window	client window including window border
app	client window excluding window border
frame	window border
title	titlebar
border	window border excluding titlebar

Table 9: Contexts for button bindings

- Next you enter your button binding in the file whose name you passed to mwm as the value for the configFile resource instance (class: ConfigFile). Enter the button binding in the file, using the syntax described under the heading "Button binding syntax" in the mwm client description.
- Specify the name of the button binding as the value for the buttonBindings resource instance (class: ButtonBindings) as described in the chapter "Using resources to customize client appearance and behavior".

You have now defined a new button binding.

## 8 Tasks for system administrators

The system administrator can perform the following tasks by graphical interaction using desktools:

<b>Task</b>	<b>Desktool</b>	<b>Description</b>
Managing login names and groups (local and global); setting preferences	User Manager	section "User management "
Managing hosts and networks	Host Manager	section "Host and network management"
Managing devices with removable media (CD-ROM, floppy disk, streamer tape); working with archives and file systems on removable media	Volume Manager	section "Managing removable volumes"
Managing, e.g. loading, installing and deleting software	Software Manager	section "Administering software"
Mounting, unmounting and sharing (parts of) file systems via NFS	NFS Manager	section "Managing the NFS"
Managing X terminals	X Terminal Manager	section "Administering X terminals"
Customizing the user environment on a system-wide or user-specific basis, e.g.	National	section "Modifying the country-specific user environment"

selecting a language or currency		
Setting the date and time	Date & Time	section "Setting the date and time"
Changing the local host mode of operation	Shutdown	section "Change the host mode of operation"
Generating or modifying tools on the Desktop	-	section "Creating objects"
Installing and managing fonts	Font Service	section "System and Diagnostic Information"

Table 10: Tasks for system administrators

The "Help" menu item on the pop-up menu of each desktool provides more information on the tasks and how to perform them.

Additionally the desktool "SysAdm" provides you with an access to the administration package OA&M. The alphanumerical user interface of OA&M is provided in a terminal emulation. You will find a detailed description of this desktool in the manual entitled "System Administration and Hardware Configuration with SYSADM".

**Note:**

The following desktools are not available on RM1000 computers: xdum, xdhm, xdvm, xdsm, xdlm, xsd

## 8.1 Creating objects

To create a tool and place it on your desktop as an icon, open the "File" menu on the main desktop and then select "Desktool" from the "New" cascade menu.

To create a toolbox and place it on the desktop as an icon, open the "File" menu on the main desktop and then select "Toolbox" from the "New" cascade menu.

You can use the "New Desktool" dialog box to define the appearance and behavior of the objects you select. If the selected object is a file type or a message type, the changes that you make will apply to all objects of this type.

You can define the image, the standard and detailed description of the object as well as the methods used to perform certain actions.

Refer to the corresponding help texts for details.



Figure 2: Creating an object

**Note:**

You must save your changes if you want them to take effect for all sessions after the current one.

For a detailed description of the XDLC configuration language, refer to the manual entitled "SINIX/windows User Environment; XDCL Desktop Configuration Language".

## 8.2 Host and network management

Desktools involved:

Host Manager	xdhm client
NFS Manager	xdnm client (Network File System Manager)
X Terminal Administration	xtadm client
Shutdown	xsd client

Table 11: Host and network management desktools

You can use these desktools to perform the following tasks:

- manage hosts
- manage the NFS
- administer X terminals
- change the host mode of operation (shutdown, restart)

Instructions on how to perform the individual tasks are given below.

### 8.2.1 Using object managers in the network

This section tells you how to start object managers on remote hosts or enable remote hosts to start object managers on a local host.

#### 8.2.1.1 Granting access from remote hosts

You can recognize object managers which you can use for managing objects on remote hosts by the fact that their icons bear the names of a specific user and host. Examples include the software and user managers.

However, the user or system administrator of the remote host must grant access to remote systems. Likewise, you or the system administrator can grant access to your system from remote hosts.

#### 8.2.1.2 Granting access to your local area

Add the name of the remote host and the login name of the remote user to your `$HOME/.rhosts` file.

If you want the remote user to be able to access your area from any remote host, replace the host name with the plus sign (+). If you want to be able to access your local area from remote hosts under your own login name, just enter the name of the remote host in the file. For example, if you add a line that contains only a plus sign to the `$HOME/.rhosts` file, you will be able to use your login name to access your local area (with the same login name) from any remote host.

#### 8.2.1.3 Granting access to the local host

If you are logged in as the system administrator, you can make the local host accessible

from specific remote hosts. In order for a remote user to be able to access an area on the local host, this local area must have the same login name as the remote area from which it is accessed.

Log in as the system administrator and add the name of the remote host from which the local host is accessed to the `/etc/hosts.equiv` file.

#### 8.2.1.4 Running managers in areas of the local host or a remote host

You can start the managers under a login name on a remote host, for example in order to access objects that are available only on that host. To do this, you must be authorized to access the remote host. section "Granting access from remote hosts" describes how to grant access to remote users.

If you want to work under login names other than your own on a local or remote host, activate the pop-up menu of the manager icon and select the entry "Define host..." or "Define user..." . When the dialog box opens, enter the host name or login name. The current parameters appear in the manager icon. If you have reset the parameters, check to make sure that you are authorized to access the required area. To do this, select "Authorize" from the icon pop-up menu.

You can restart the object manager for each valid location for which you have access authorization, which means that you can have the main windows of several different managers open at the same time. However, you can call each location only once.

#### 8.2.2 Managing hosts

The "Host Manager" desktool enables you to display the hosts and networks which your host recognizes.

Under certain conditions, you can manage hosts and networks as objects. The host on which the host manager is running must be either the network management host for the programmed NIS domain, or the network management services must be deactivated (NIS). The host manager uses the `/etc/hosts` and `/etc/networks` system files to obtain information on hosts and networks. The "Hosts" and "Networks" objects are displayed as icons on a host manager window. By default, this type of window displays all hosts available on your system. However, it is also possible to display only those hosts belonging to a certain network.

In system manager mode, you can

- change host and network attributes
- delete hosts and networks permanently
- place hosts and networks in the trash
- enter new hosts and networks and
- assign to the hosts and networks icons which can also be used by other object managers

To do this, the host on which the host manager is running must be either the network management host for the programmed NIS domain, or the network management services must be deactivated (NIS).

You can also view and modify the NIS attributes of the system on which the host manager is running.

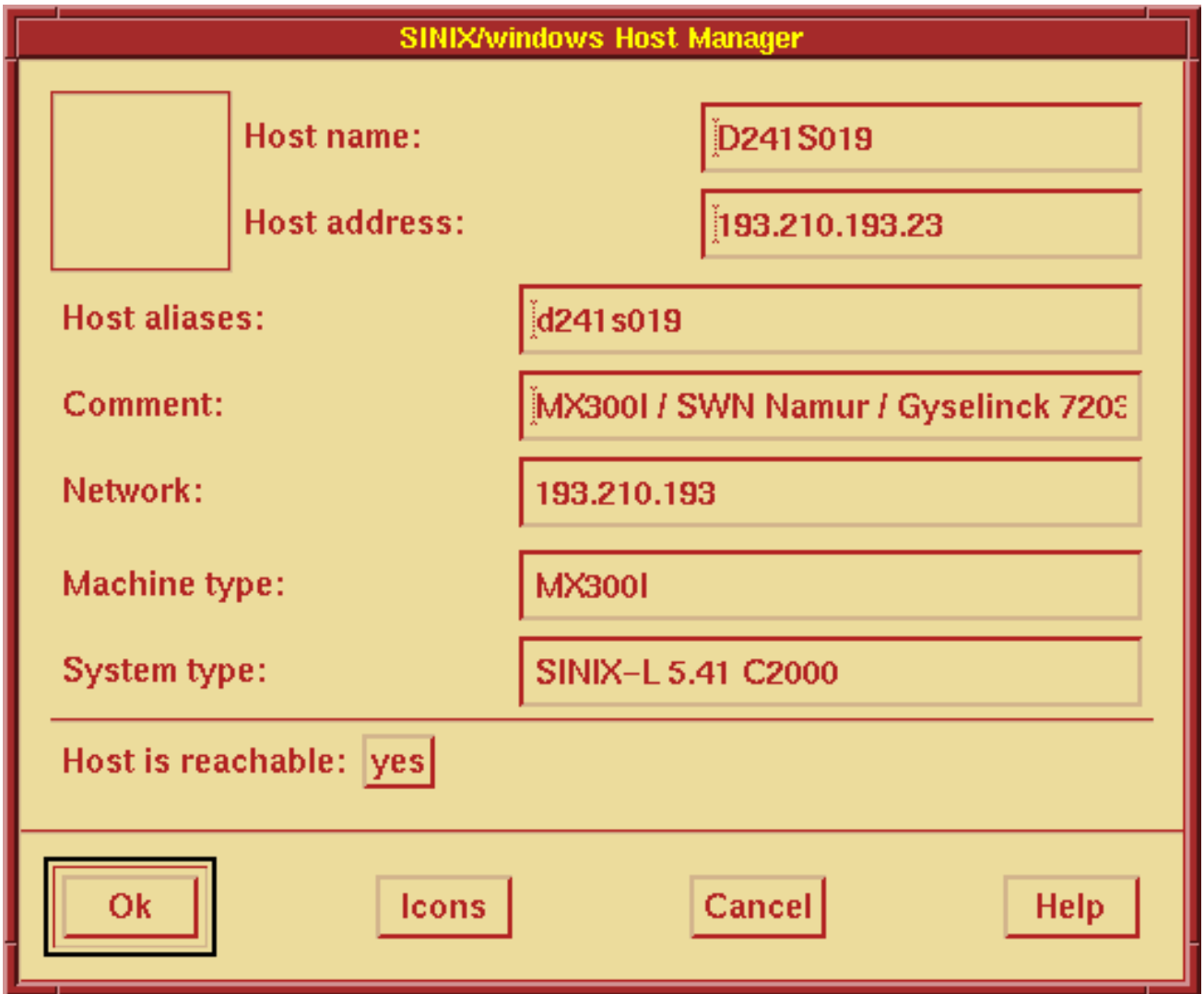


Figure 3: Host manager

The table below shows functions of the controls in the "Host Manager" dialog box:

"Host name" and "Host address"	These input fields show the host names and their Internet addresses
"Host aliases" and "Comment"	The "Host aliases" field can define alternative names for the host, and the "Comment" field can contain additional information about the host.
"Machine type" and "System type"	The "Machine type" field displays the name of the computer type, and "System type" shows the name and version of the operating system.

	This information cannot be determined for some computer types. In this case, the fields remain blank.
"Host is reachable" pushbutton	The pushbutton labeled "Host is reachable" indicates whether the host is available from the network. This state can change. Press the pushbutton to update the status information. This query can take several seconds.

Table 12: Controls in the host manager dialog box

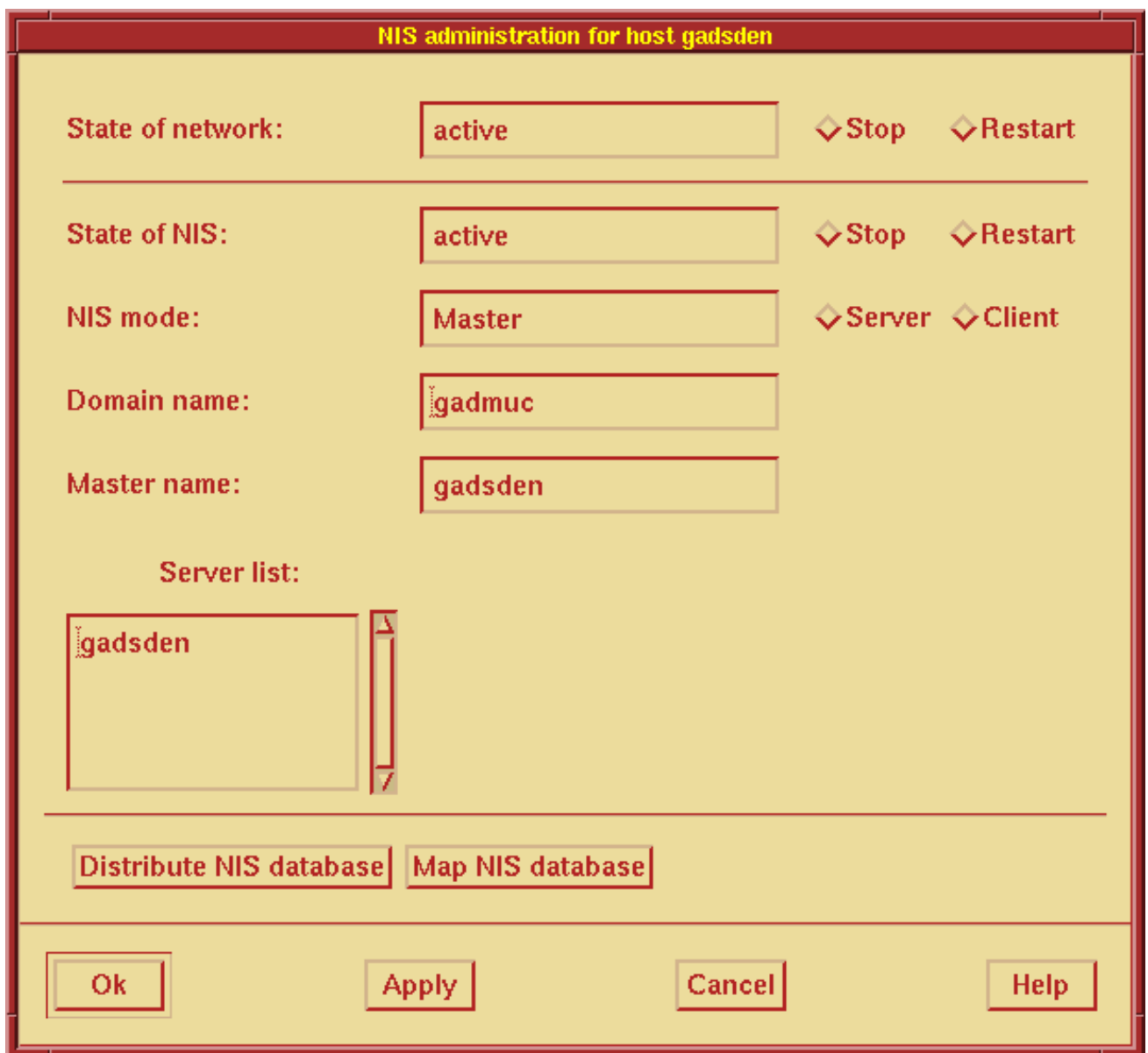


Figure 4: NIS Administration

The NIS Administration window is a dialog box which shows the status of the local host in the network along with the NIS attributes. The displayed values can change, depending on various requirements.

Below is a description of the controls in the "NIS Administration" dialog box:

State of network	Shows the status of the network software on the local host.	
	not installed	The network software is not installed.
	not configured	The local host has not yet been configured for network operation.
	inactive	The local host is not mounted in the network.
	inconsistent	The local host is mounted in the network, but its status is inconsistent. Remount the host in the network.
	active	The local host is mounted in the network and its status is OK.
	The "Stop", "Start" and "Restart" pushbuttons are displayed, depending on the status	Always stop the network software only on the local host and not from a device (monitor) connected to the host via the network. The host is unmounted when you stop the network software, even if you shut down the system and restart it again. The host is mounted back in the network only when you run the command "Restart".
State of NIS	The status of the NIS software on the local host is displayed.	
	not installed	The NIS software is not installed.
	inactive	The NIS services are not being used to administer the local host.
	inconsistent	The local host is being administered by the NIS services, but its status is inconsistent. Restart the NIS software.
	active	The local host is being administered by the NIS services. Its status is OK.
	The "Stop", "Start" and "Restart"	The NIS services remain stopped until you restart them with the "Start" command. Restarting the

	pushbuttons are displayed, depending on the status	host does not activate the NIS services.
NIS mode	This shows the current NIS mode of the local host.	
	Master	The local host is the network management host on which the host and network definitions of the assigned NIS domain, among other things, are managed. The definitions are distributed to all hosts mounted in this domain. You can change the administered host and network data, but you should note that this data applies to the entire NIS domain.
	"Master" pushbutton	This button makes the local host the network management host for the assigned NIS domain. In the case of new domains for which no network management host yet exists, you must set the NIS mode to "Master" before you can start the NIS services. If you want the local host to be the network management host for an existing domain, you must enter its name in the list of deputy hosts (server list) on the current network management host. After changing the NIS mode to "Master", you should make the previous network management host the deputy (using the "Server" pushbutton). If there are additional deputies, they must be associated with the new management host (using the "Master name" field).
	Server	The local host is a deputy network management host for the assigned NIS domain. It is entered in the deputy list (list of servers) on the network management host. It regularly retrieves NIS data from the management host in order to provide it to client hosts..

	"Server" pushbutton	This pushbutton makes the local host the deputy host for the assigned NIS domain. To switch the NIS mode to "Server", the local host must be entered in the deputy list (list of servers) on the network management host. If the local host is the network management host, you must first make another host the management host before you can make the local one the deputy (using the "Master" pushbutton).
	Client	The local host does not have a network management function. It regularly retrieves NIS data from the deputy or network management host of the assigned NIS domain.
	"Client" pushbutton	This pushbutton makes the local host the client host of the assigned NIS domain. If it was previously configured as the network management host, you must first make another host the management host before you can change the local one to a client host (using the "Master" pushbutton). If the local host was previously the deputy or network management host, you must remove its name from the list of deputies.
Domain name	The name of the NIS domain to which the local host is assigned. If the host manager is in system manager mode, you can assign the local host to a different domain.	
Master name	This field displays the name of the network management host for the domain to which the local host is assigned. If the host manager is in system manager mode, the NIS software is active and the local host is an NIS server, you can change the master name.	
Auto bind mode	If the local host is an NIS client, this field shows the auto bind mode status. If the host manager is in system manager mode, you	

	can change the auto bind mode status.	
	On	"On" means that the binding server list is automatically customized to the list of current NIS servers. The current entries in the binding server list are ignored. You cannot change the binding server list when auto-bind mode is set to "On". When you press the pushbutton to change auto bind mode, you can edit the binding server list without having to press the "Apply" pushbutton. If you press the pushbutton again, i.e. to change back to the old status, all changes to the binding server list will be discarded.
	Off	When the host manager is in system manager mode and auto bind mode is set to "Off", you can change the binding server list.
Server list	<p>This field shows a list of the deputy servers assigned to the NIS domain, provided that the NIS software is active or the local host is a network management host.</p> <p>If the host manager is in system manager mode, and the local host is a network management host, you can change the server list.</p> <p>When you change the data on the network management host, all deputy hosts included in this list automatically receive a copy of the latest management data. At startup each active host in this domain is bound to any of these deputy hosts so that it can request the configuration data from this deputy. If a deputy host fails, the host bound to it is rebound to avoid access problems. Depending on the architecture of the LAN involved, a practical number of deputy hosts must be set up.</p>	
Bind server list	<p>If the local host is an NIS client, this field shows the list of NIS servers which try to obtain information from NIS files via the local host as the need arises. We refer to this as "binding to the server".</p> <p>The local host binds to the first server in the list from which it receives a reply. The sequence of servers in the list is therefore very important.</p>	
	*	The local host sends a request to all hosts in the assigned NIS

		domain and binds to the first server that replies.
	+	The local host sends a request to all hosts in the assigned NIS domain and binds to the first server that replies and that is included in the list following the "+" sign.
SysAdmin		The "SysAdmin" pushbutton is displayed if the system administrator did not already invoke the host manager.
Map NIS databases		This pushbutton is always displayed when NIS is active. The "Map NIS databases" pushbutton is displayed in order to program the system as described in the network management files.
Fetch NIS databases		<p>If the local host is an NIS server and NIS is active, the "Fetch NIS databases" pushbutton is displayed in order to retrieve the network management files for the current domain from the network management host. By doing this, you ensure that the network management system remains consistent.</p> <p>This data is normally transferred automatically when you make changes to the network management files on the network management host and when you restart the system on your host.</p> <p>This function is carried out automatically at certain intervals (usually once every hour) and each time the system is started. As a result, you generally have to run this function only when you need the latest data right away.</p>
"Distribute NIS databases" push button		<p>If the local host is a network management host and NIS is active, the "Distribute NIS databases" pushbutton is displayed so that you can start distributing the network management files for the current domain to the NIS servers.</p> <p>As a rule, an automated procedure is used to send the modified data on the network management host to the defined deputy hosts. The deputy hosts, in turn, request modified data upon system startup and at regular intervals.</p> <p>If the connection to a deputy host is interrupted for any reason, and if this host is to receive prior changes immediately, it is both useful and necessary to transfer the data with this function.</p>
"Apply" pushbutton		The "Apply" pushbutton accepts all modified values. In other words, the changes are carried out. Unlike the "OK" pushbutton, you do not leave the NIS Administration window but can continue making

other changes.

Table 13: Controls in the NIS Administration dialog box

### 8.2.3 Managing the NFS

The "NFS Manager" desktool enables you to display information about file systems and file system components which are either mounted on you host via NFS (Network File System) or are provided to remote hosts via NFS. System administrators can administer the mount points for mounting remote file systems and the share points that are activated for mounting local file systems in remote ones in the form of objects:

The NFS manager uses the `/etc/vfstab` and `/etc/dfs/dfstab` system files to obtain information about mount and share points.

The "share points" and "mount points" objects are displayed as icons in an NFS manager window. Share points appear in upper window panes and mount points in lower window panes:

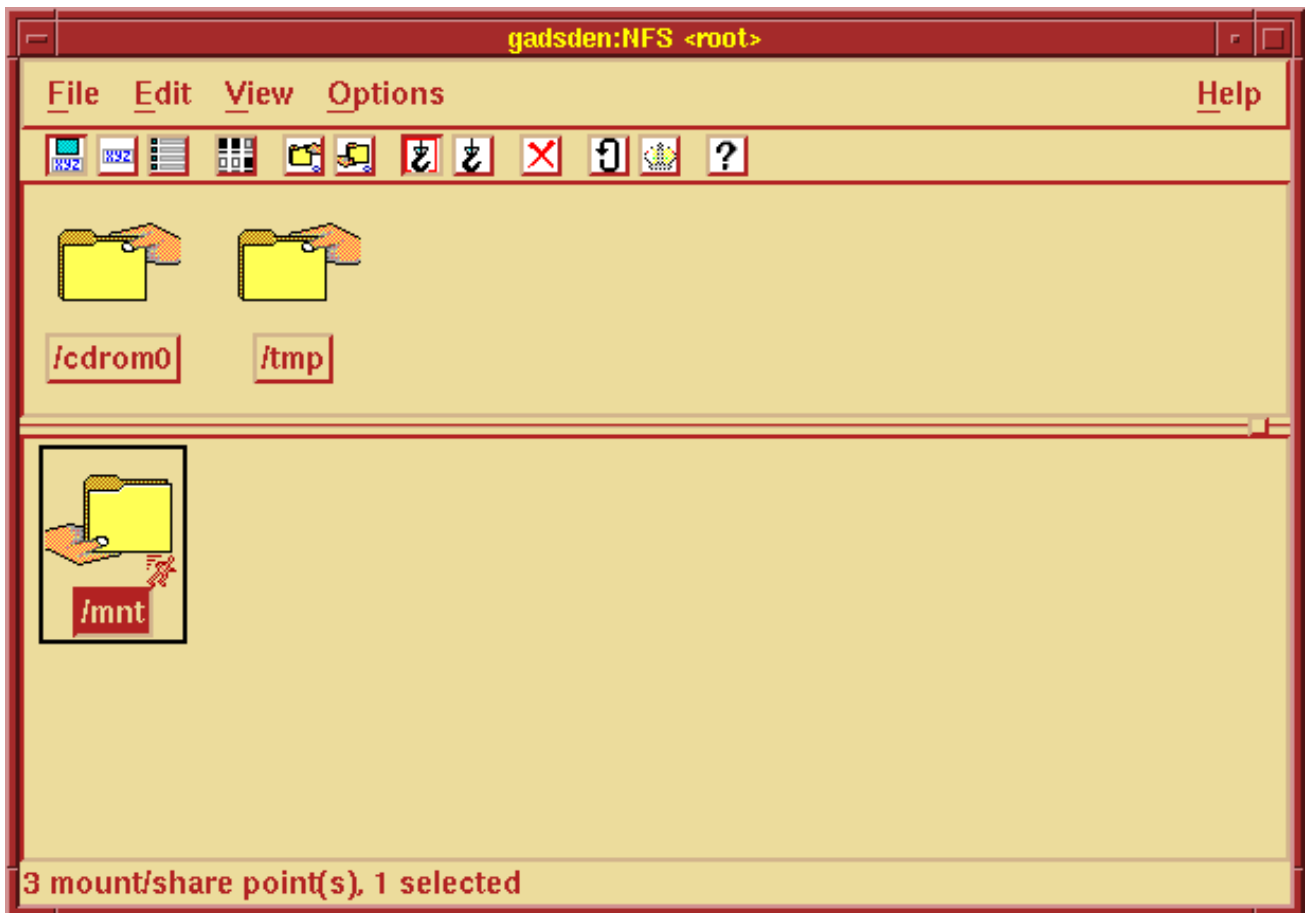


Figure 5: NFS manager

#### 8.2.3.1 Managing mount points as objects

System administrators can mount files systems from remote hosts in a file system on the local host via NFS. They can change the attributes of a file system mounted via NFS or

remove file systems from the list of mounted file systems.

The lower window pane displays the mount points which can be mounted with the auto-mounter, provided that the latter is available on the local system. Each mount point has a local and a remote name along with other attributes, The local name is an absolute path name in the local file system. The remote name is the absolute path name in the remote file system, preceded by the name of the remote host.

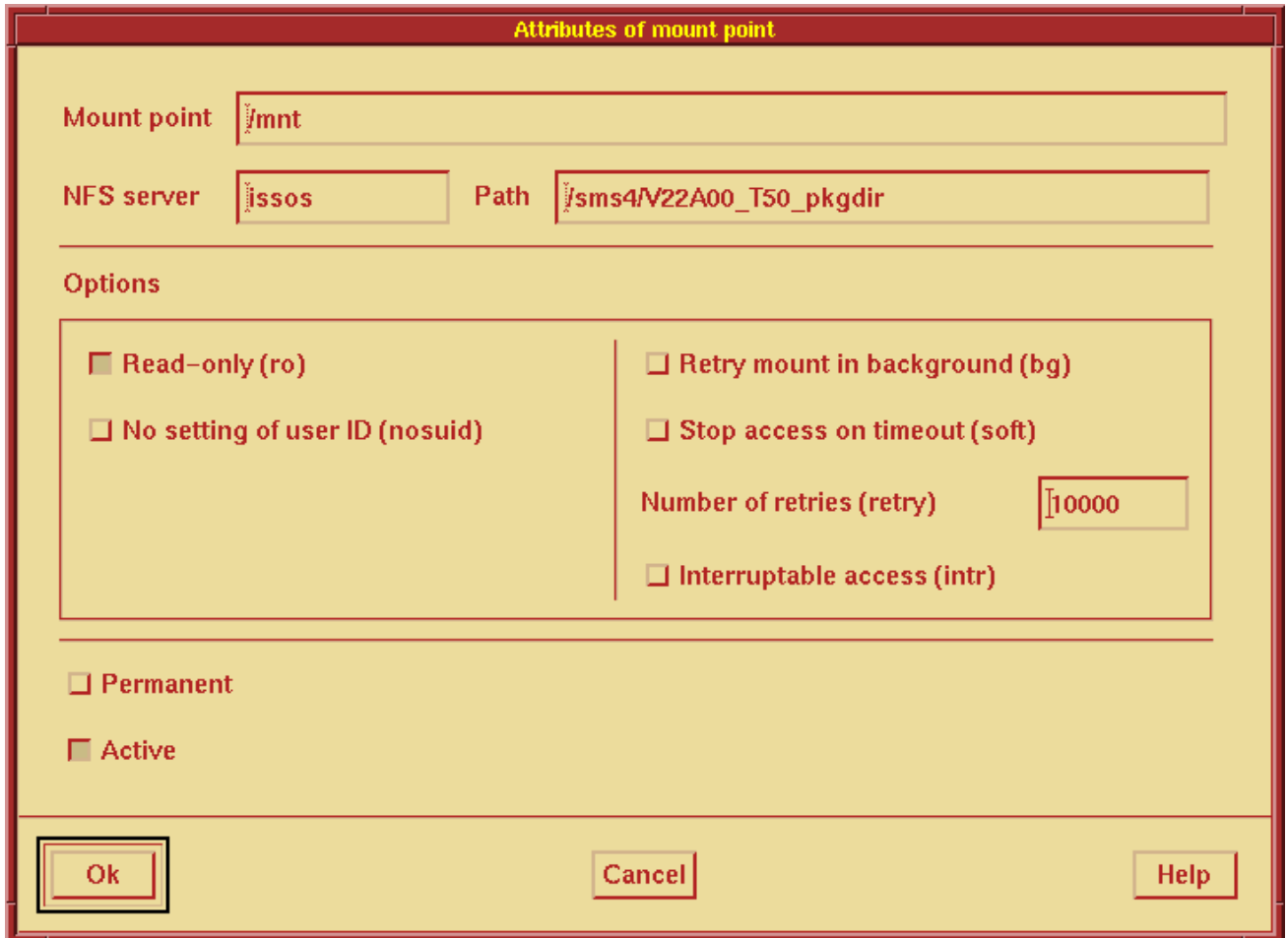


Figure 6: "Attributes of mount point" dialog box

This dialog box displays the attributes for a mount point. You can change the attributes when the NFS manager is running in system manager mode. You have to fill out only the "NFS server", "Path" and "Mount point" fields.

The following table describes the individual controls:

<p>"NFS server" and "Path"</p>	<p>These input fields display the remote name of the mount point. The remote name consists of two parts: the absolute path name in the remote file system and the name of the remote host from which the file system or file system component was provided. The remote host is referred to as the NFS server, since it provides the local host with a file system (component) via NFS.</p>
--------------------------------	--

"Mount point"	This input field displays the local name of the mount point. The local name is the absolute path name in the local file system. The local host is referred to as the NFS client.	
"Options"	This selection box contains the mount point attributes which define access from remote hosts.	
	Read-only	If this option button is selected, the files and directories under the mount point are in read-only mode. Otherwise the NFS determines them to be read/write files and directories. If "Read/write" mode is specified for a mount point in the system file, although the NFS is granted read-only access at the moment, the mount point is displayed twice: once as a permanent but inactive mount point with read/write access and again as a temporary but active mount point with read-only access.
	No setting of user ID	If this option button is selected, you cannot set a different user ID for working with files or directories under the mount point (no s-bit set for user). Otherwise this is possible.
	Retry mount in background	If this check option button is selected, then next and any additional mount attempts are carried out in the background. The maximum number of mount attempts is displayed in the "Number of retries" input field.
	Number of retries	This input field displays the maximum number of mount attempts.
	Stop access on timeout	If this option button is selected, the attempt to access a file system via NFS is terminated with an error message after a timeout. Otherwise a warning is issued and the access attempts continue.
	Interruptable access	If this option button is selected, an access attempt to which the NFS server fails to respond is terminated with interrupt signals. Otherwise this does not occur.
"Permanent" option button	The function of this option button depends on whether you are merely displaying attributes in this dialog box or whether you can modify them as well.	

	Show	If the option button is already selected when you open the dialog box, the mount point is entered in the <code>/etc/vfstab</code> system file. In this case, it is provided automatically each time you start the system. Otherwise it is not provided.
	Modify	If this option button is selected, the mount point is entered in the <code>/etc/vfstab</code> system file along with the displayed attributes (which may be modified) as soon as you press the "OK" button. Otherwise it is not entered in the file.
"Active" option button	The function of this option button depends on whether you are merely displaying attributes in this dialog box or whether you can modify them as well.	
	Show	If the option button is already selected when you open the dialog box, the mount point is active. This means that files and directories under this mount point on the NFS server are available. Otherwise the mount point is inactive and the files and directories under the mount point on the NFS server are unavailable.
	Modify	If this option button is selected, the mount point is activated as soon as you press the "OK" pushbutton. Otherwise the mount point is deactivated.

Table 14: Controls in the "Attributes of mount point" dialog box

**Warning:**

If you press the "OK" pushbutton without selecting the "Permanent" and "Active" option buttons, the mount point will be deleted.

**8.2.3.2 Managing share points as objects**

System managers can provide file systems on the local host to remote hosts via NFS and define the file system attributes. These file systems can be mounted on approved remote hosts via NFS.

Each share point has a name - its absolute path name in the local file system - and other attributes.

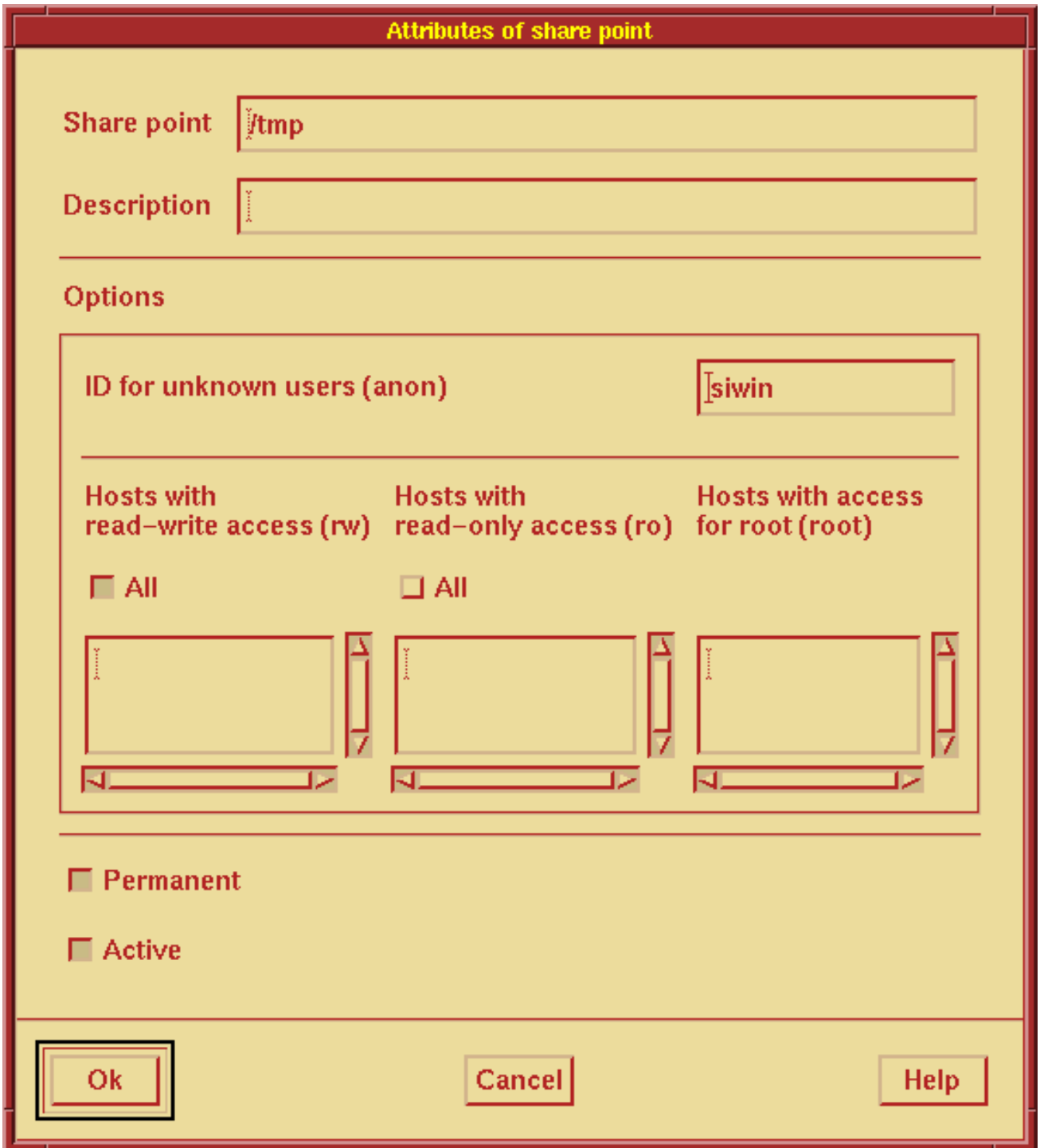


Figure 7: "Attributes of share point" dialog box

This dialog box displays the attributes of a share point. You can modify the attributes if the NFS manager is in system manager mode. You have to fill out only the "share point" field.

The table below shows the functions of the individual controls

:

"Share	These input fields shows the absolute path
--------	--

<p>point" and "Description"</p>	<p>name of the share point and a remark describing the share point. The file system or directory activated for mounting should not be located under a file system or directory which NFS has mounted. However, a file system or directory can be mounted under the share point via NFS.</p>	
<p>"Options"</p>	<p>This selection box contains the attributes of a share point which define access from remote hosts.</p>	
	<p>ID for unknown users</p>	<p>This input field shows the user ID assigned to the "root" user of a remote host if the remote host is not entered in the "Hosts with access for root" list. If no user ID is displayed, the effective ID UID_NOBODY is assigned to unknown users. If user ID "1" is displayed, access is denied.</p>
	<p>Hosts with read-write access</p>	<p>If the "All" option button is selected, the read-write share point is provided to all hosts which send NFS requests to the local host. Otherwise the share point is provided only to those hosts named in the list beneath this button. If a host is entered in both the "read-write" and "read-only" lists, this entry overrides the summary entry in the opposite list when you select the "All" option button. For example, if read and write access is granted to all hosts, but the host named host is entered only in the "Nur Lesen" list, host is indeed granted just read-only access.</p>
	<p>Hosts with read-only access</p>	<p>If the "All" option button is selected, the read-only share point is provided to all hosts which send NFS requests to the local host. Otherwise the share point is provided only to</p>

		those hosts named in the list beneath this button.
	Hosts with access for root	Only the root users of the hosts specified in the list beneath this button can use the share point under the "root" user ID.
"Permanent" option button	The function of this option button depends on whether you are merely displaying attributes in this dialog box or whether you can modify them as well.	
	Show	If this option button is already selected when you open the dialog box, the share point is entered in the <code>/etc/vfstab</code> system file. In this case, it is provided automatically each time you start the system. Otherwise it is not provided.
	Modify	If this option button is selected, the share point is entered in the <code>/etc/vfstab</code> system file along with the displayed attributes (which may be modified) as soon as you press the "OK" button. Otherwise it is not entered in the file.
"Active" option button	The function of this option button depends on whether you are merely displaying attributes in this dialog box or whether you can modify them as well.	
	Show	If this option button is already selected when you open the dialog box, the share point is active. This means that remote hosts can access files and directories under this share point and mount them in its local file system according to the other share point attributes. Otherwise the share point is inactive and the files and directories are unavailable to remote hosts
	Modify	If this option button is selected, the share point is activated as

		soon as you press the "OK" pushbutton. Otherwise the share point is deactivated.
--	--	--

Table 15: Controls in the "Attributes of share point" dialog box

**Warning:**

If neither the "Permanent" nor the "Active" option button is selected when you press the "OK" pushbutton, the share point will be deleted.

**8.2.4 Administering X terminals**

**Note:**

Before you begin configuring X terminals, make sure that the shell command `/usr/bin/X11/xtadmppriv ID Yes` is listed for the login name under which you want to configure the X terminals. You can cancel access authorization again using the shell command `/usr/bin/X11/xtadmppriv ID No`.

The "X Terminal Administration" desktool enables you to administer a large number of X terminals in a network. It facilitates the initial startup and configuration tasks for X terminals, the transfer of X terminals to another boot server host and the storage of redundant data for the X terminals to be administered. All necessary data is stored in a database. All other files and commands derive their information for operating X terminals from this database. Use the table editor to view and modify the administration configuration data. The pop-up menu for this desktool contains the following items:

Edit type&host table	Starts an editor for the X terminal type table
Update type&host setup	Updates the X terminal type table.
Check type&host setup	Checks the consistency of the X terminal type table.
Edit configuration table	Starts an editor for the configuration table.
Update configuration setup	Updates the configuration table.
Check configuration setup	Checks the consistency of the configuration table.
Save setups	Saves the current settings.
Restore setups	Restores the settings last saved.

Table 16: X Terminal Administration

The individual elements in the type and host table as well as the configuration table are documented in the help texts.

### **8.2.5 Change the host mode of operation**

The "Shutdown" desktool enables you to change the mode of operation of a host. It may be necessary to switch to a different mode if the system administrator wants to perform data backups, carry out administrative tasks or shut down the host.

#### **Warning:**

When you change a host's mode of operation, this will affect all users still working on the host. All current processes are terminated when you shut down the host. You should use the power switch or power plug to shut down the host only in emergencies (e.g. if lives are in serious jeopardy). Processes which must be terminated in the usual way are always running on a SINIX host. If you do not terminate these processes properly, data may be lost.

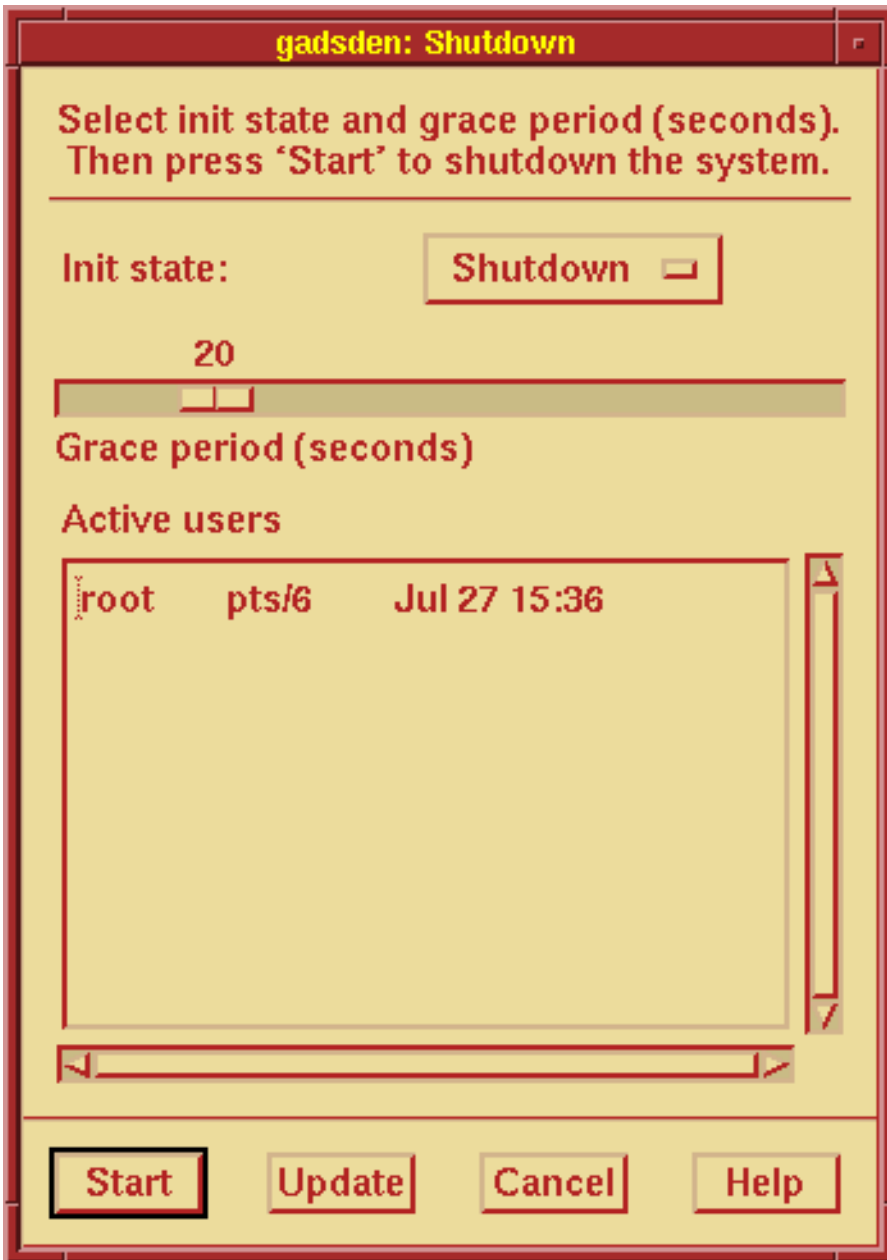


Figure 8: Shutting down the host

The table below lists the possible modes of operation (init status):

Shutdown	This mode terminates all processes, enabling the power to be turned off. The power is cut off automatically, depending on the host firmware.
Restart	This mode terminates all processes and then reloads the system with the status defined in the <code>initdefault</code> entry in the <code>/etc/inittab</code> file.

Single-user mode	Stops all processing that are running with a terminal assignment. Processes that are not associated with a terminal (daemon processes) are not stopped. File systems are not mounted and software is not started. Only one user can work on the system after being requested for a password on the console.
------------------	---

Table 17: Modes of operation

You can use the "Grace period" slider to set a delay time in seconds after which the mode changes. You can select a range from 0 to 120 seconds for the delay time; the default setting is 20 seconds.

The "Active users" scroll list provides an overview of the users who are logged in. These users receive a message when the mode of operation changes so that they can log out in time.

**Note:**

It is not advisable to change a host's mode of operation while users are still logged in. If necessary, give users sufficient prior notice of a system shutdown, for example by e-mail.

If you want to change a host's mode of operation while users are still logged in, be sure to make the grace period long enough so that the users can log out properly.

To start changing the mode of operation, press the "Start" button.

To terminate the client without changing the mode of operation, press the "Cancel" button. You can not cancel the process once you have pressed the "Start" button.

### 8.3 User management

Desktool involved

:

User Manager	xdum client
--------------	-------------

Table 18: User Manager desktool

This desktool enables you to perform the following tasks:

- retrieve information on users and groups
- add login names and user groups
- change login names
- delete login names

Instructions on how to perform the individual tasks are given below.

#### 8.3.1 Retrieving information on users and groups

After you start the user manager, an overview window appears showing all users for whom a login name has been set up on the local host. If you want to retrieve detailed information on a login name, double-click on the appropriate icon.

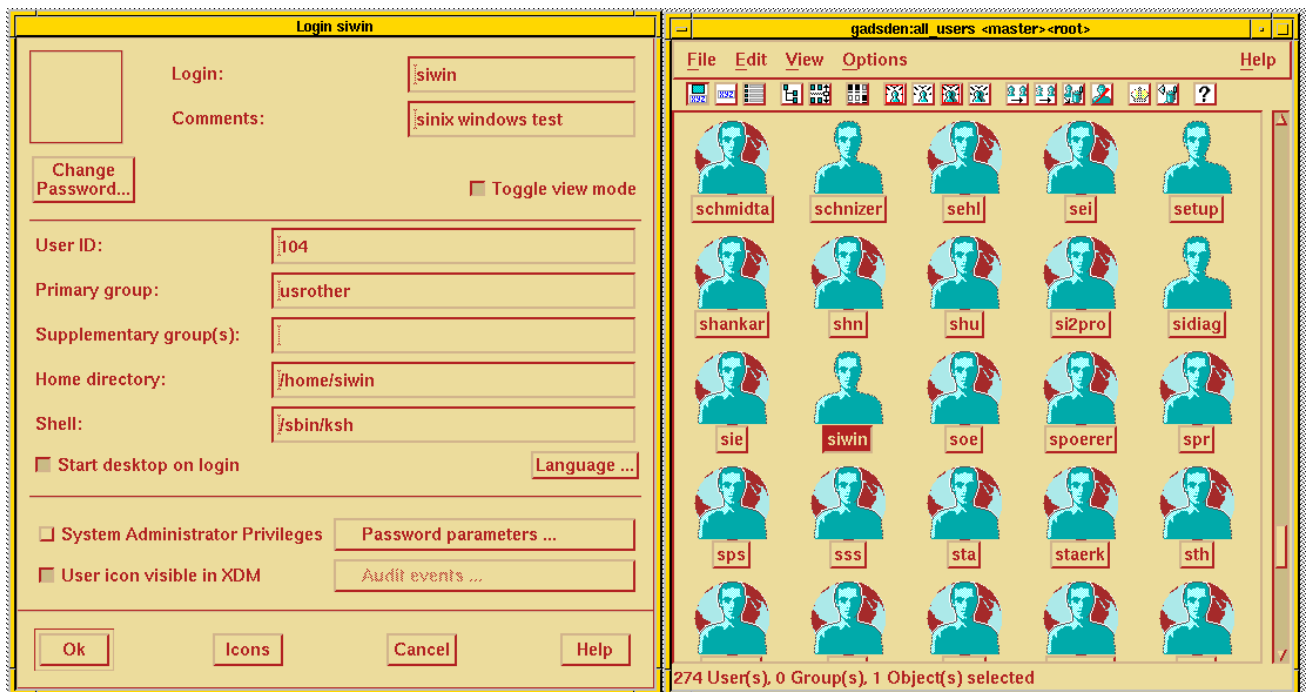


Figure 9: User manager and dialog box for a login name

If you want to retrieve detailed information about more than one user, open the "View" menu item and then the "View as" cascade menu and click on the "Extended" item. Alternatively, you can click on the corresponding icon on the icon bar.

Account Name	User ID (UID)	Group ID (GID)	User-specific Comment	HOME Directory	Login Shell	NIS User Category
si2pro	2188	2010	J. Carlo Huber-Pongratz,HESW413,49921,,universe(att)	/usr/si2pro	/bin/ksh	global
sidiag	87	1	Unprivileged Field Service User	/home	/sbin/sh	local
sie	2009	2010	Bernhard Siebert,PD123,47693,,universe(att)	/usr/sie	/bin/sh	global
siwin	104	100	sinix windows test	/home/siwin	/sbin/ksh	local
soe	2106	2010	Josef Schoellhorn,PD242,2843,,universe(att)	/usr/soe	/bin/csh	global
spoerer	2005	2010	Gerhard Spoerer,HESW412,40618,,universe(ucb)	/usr/spoerer	/bin/sh	global
spr	2179	2010	Simon Roach,CON,47320,PD243,universe(att)	/usr/simon	/bin/sh	global
sps	2416	2010	Matthias Roessler,PD241,40907,,universe(att)	/usr/sps	/bin/ksh	global
sss	2076	2010	Konrad Sassen,PD241,49027,,	/usr/sss	/bin/csh	global
sta	2340	2010	Stefan Arntzen,PD243,42019,,universe(att)	/usr/sta	/bin/sh	global
staerk	2206	2010	Reinhard Staerk,PD443,2659,,universe(att)	/usr/staerk	/bin/sh	global

Figure 10: User manager with overview table

This overview table provides the following information on each login name:

- account name
- user ID (UID)
- group ID (GID)
- user-specific comment
- HOME directory of the corresponding account
- login shell
- NIS user category

section "Adding login names and user groups" will tell you how to interpret and, if necessary, modify these parameters.

### 8.3.2 Adding login names and user groups

If you want to create a new login name with a new user group on your system, you must first create the user group and then the login name.

#### 8.3.2.1 Creating a user group

To create a new user group, select "File" from the menu bar; then open the "New" and "Group" cascade menus consecutively and click on either "Local" or "Global".

If you choose, "Local", you will create a local user group; if you choose "Global", you will create a global user group. You can set up a global user group only on the network management host of the NIS domain.



Figure 11: Creating a user group

The table below shows the functions of the input fields in the dialog box :

Group name	Enter the name of the new group in this field. The name must be unique within the system.
Group ID	In this field, enter the identification number (GID) which you want to assign to the new group. The GID clearly identifies the new group on the system. By default, the field displays the next available GID for this number.
Primary members	In this field, enter the list containing all the users for whom the displayed group is to be the primary group. The primary group is the group that is active when the user logs in.
Supplementary members	In this field, enter all the users for whom the displayed group is to be the secondary group. The secondary group is the group which users can make the active one with the <code>newgrp</code> command when they log in .

Table 19: Input fields in the "New group" dialog box

8.3.2.2 Creating login names

When you create a new user, the system uses the default settings which you defined in the "User Defaults" input window. You can change these settings for individual users when you click on the "Local" or "Global" item in the "File'/New'/User" cascade menu. A global login name can be created only on the network management host of the NIS domain.

The screenshot shows a dialog box titled "New login" with a yellow background and a red border. The dialog is organized into several sections:

- Top Section:** Contains a "Login:" label with an empty text input field, a "Comments:" label with an empty text input field, a "Change Password..." button, and a "Toggle view mode" checkbox.
- User Information Section:** A series of labels and text input fields: "User ID:" with "792", "Primary group:" with "usrother", "Supplementary group(s):" with an empty field, "Home directory:" with "/home/", and "Shell:" with "/sbin/sh".
- Options Section:** Includes a "Start desktop on login" checkbox, a "Language ..." button, a "System Administrator Privileges" checkbox, a "User icon visible in XDM" checkbox, a "Password parameters ..." button, and an "Audit events ..." button.
- Bottom Section:** Contains four buttons: "Ok", "Icons", "Cancel", and "Help".

Figure 12: Creating users

The table below shows the functions of the input fields in the dialog box :

Login	Enter the login name here. A user logs in under this name. This name must be unique throughout the system/
Comments	Enter a comment for the login name in this field. Useful information includes the user's name, organizational unit and telephone number.
User ID	In this field, enter the identification number (UID) which you want to assign to the new login name. The UID clearly identifies the new user in the system. By default, the field displays the next available UID for this number.
Primary group	The primary group is the group to which a user is assigned after he logs in.
Supplement ary group	In this field enter the groups to which you want the user to belong in addition to the primary group. After he logs in, the user can activate the newgrp command for each of these groups.
Home directory	This is the name of the directory where the user stores his files. If is defined as a directory which the user can reach by typing the <code>cd</code> command without an argument.
Shell	In this field enter the name of an existing shell in the system. Click with the right mouse button to open a pop-up menu containing the shell selection options.
Start desktop on login	In this field you specify whether you want the SINIX/windows desktop to start up when the user logs in.
Language	This starts the "International" desktool. For more information, refer to section "Modifying the country-specific user environment".

Table 20: Input fields in the "New login" dialog box

### 8.3.2.3 Security precautions

You can use certain mechanisms to protect, monitor and disable each login name or provide it with system administrator privileges. You specify these things in the "Password

parameters" dialog box.

SystemAdminis trator Privileges	In this field, you specify whether the user can perform system administrator activities with the <code>sysadm</code> command or the "Sysadm" desktool. In addition, it is not necessary to enter a password when switching a desktool to administrator mode	
User icon visible in XDM	In this field you specify whether the icon representing the login name appears in the login window.	
Password parameters	Maximum days between password changes	In this field enter the minimum number of days which must pass before the password can be changed.
	Minimum days the password is valid	In this field enter the maximum length of time the password can be valid.
	Number of days fopr warning message	In this field enter the number of days prior to the expiration of a login name when a message is displayed indicating that the password is about to become invalid.
	Login inactivity	In this field enter the number of days the login name can remain unused before it is disabled.
	Login expiration date	In this field specify the last day on which the user can log in before his login name is disabled. The date must be in the

		future and can be entered in the format mm/dd/yy or month dd, yyyy or in the country-specific format (e.g. mm.dd.yy).
	Password status	In this field specify whether a password is assigned to the login name, whether this password has been disabled or whether the user is forced to define a new password when he logs in.
Audit events	If the AUDIT product is not installed on the system, this field is grayed out. For more information about the procedure for logging system events, refer to the "AUDIT System Administrator Guide".	

Table 21: Activating security mechanisms

### 8.3.3 Changing login names

It may be necessary to change a login name, for example, when a user wants to use a different shell or a login name is to be transferred to a different area on the hard disk.

#### Warning:

Make sure that the user concerned is not logged in.

To change the parameters of a login name, click on the corresponding icon. The dialog window contains the controls described in section "Adding login names and user groups".

### 8.3.4 Deleting login names

A login name that is no longer being used poses a security risk, which is why unused login names must be deleted.

#### Warning:

Before you delete a login name, make sure that the corresponding user is not logged in. If necessary, save any login name data that is still needed.

Before you delete a user group, make sure that no login names are assigned to the user

group you want to delete.

To delete a login name, click on the corresponding icon and select "Delete" from the "Edit" pull-down menu.

## 8.4 System and Diagnostic Information

Desktools involved

:

Information	Client moinfo (Motif Information)
System-Info	Client xsi (System Information)
Log Browser	Client xdlm (Log Manager)

Table 22: Desktools for obtaining diagnostic and system information

This desktool enables you to perform the following tasks:

- retrieve information on the system
- retrieve an overview on the system
- retrieve log messages

Instructions on how to perform the individual tasks are given below.

### 8.4.1 Retrieve System Information

Using the Information desktool you can obtain the latest data on the following topics:

- Host
- Display
- Disk space
- Processes

This window presents the information you previously requested by selecting an item in the pull-down menu.

How the output is presented depends on the type of information. You can, if allowed, switch the kind of presentation by selecting 'Standard Icons' or 'Extended':

System Information	General information	This function provides you with general information on the system on which you are logged in.
	Display information	This function provides you with information on the display you are currently using.
Disk Space	Free storage space in the file systems	This item gives you information on the disk usage of the various file systems.
	Space occupied by own document	This item provides information on the amount of disk space occupied by documents under your home directory.

	s	
Active processes	All processes	This entry provides information on all the processes currently active on your system.
	Own processes	This entry provides information on all active processes associated with your login name.
	Processes on display	This entry provides information on all the processes currently associated with your display.

Table 23: Elements of the pull-down menu 'Information'

#### 8.4.2 System Overview

The desktool system information allows you to view important information about the host.

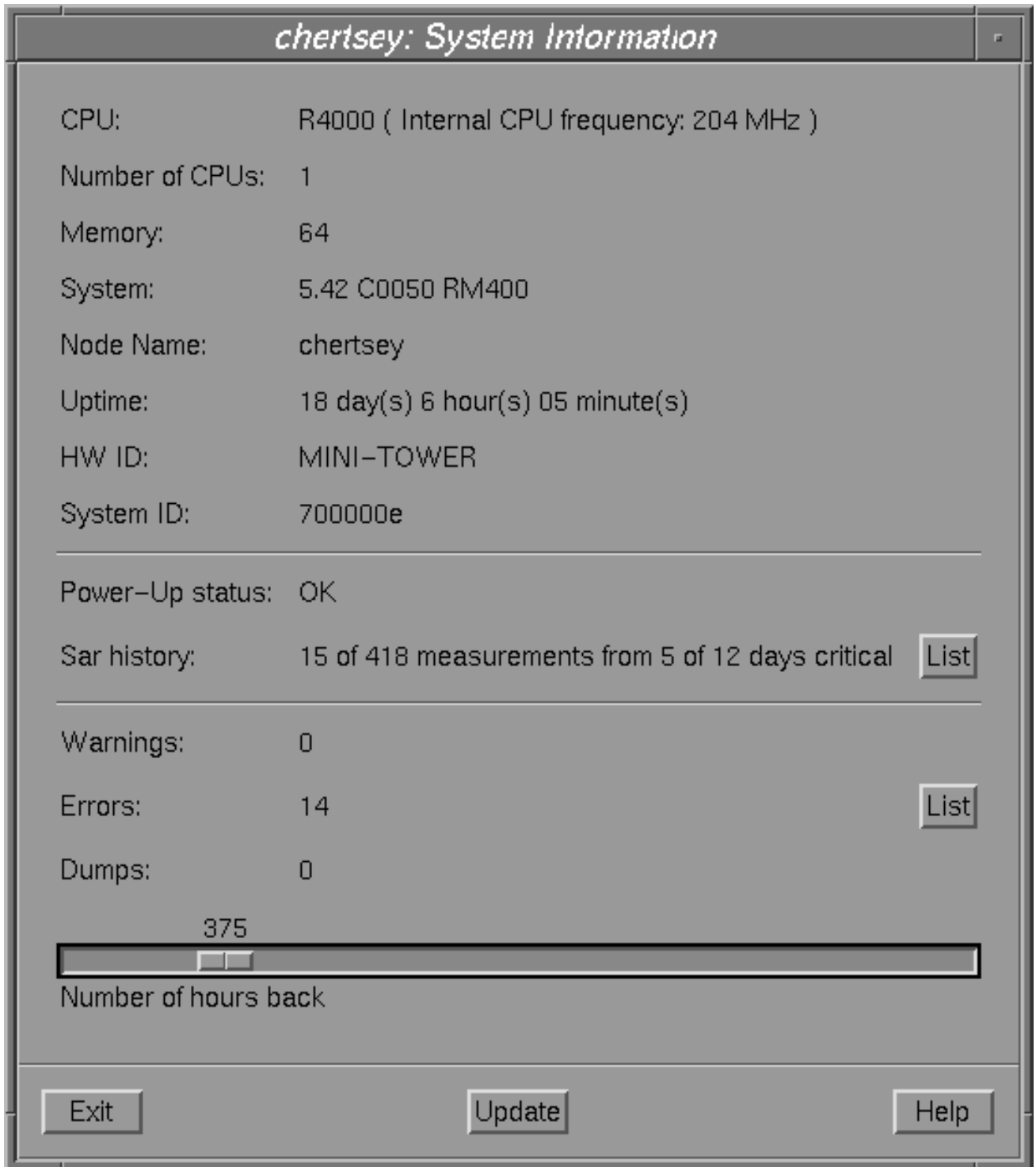


Figure 13: System Information

The table below shows the function of the individual controls:

CPU	Processor name of the host
number of	Number of processors in the host

CPU	
Memory	Size of memory (in MB)
System	Operating system version and host type
Node Name	Node name of the operating system
Uptime	Time elapsed since last system start
HW ID	Hardware-specific serial number of the host
System ID	License number of the system
Power-Up status	Indication of whether problems have occurred in the power-up tests of the host. You can view any error detected by pressing the 'List' push button.
Sar history	Indication of whether possible performance problems have been detected in the system activity history. The following displays are possible:
	OK No performance problems have been identified.
	No determination possible No history available
	<x> of <y> measurements on <n> of <m> days
	x number of measurements indicating possible performance problems
	y total number of available measurements
	n number of days on which such problematic measurements have occurred
	m total number of days in history
	For further information refer to section "Sar History"
Warnings	Number of warning-type log messages (level 4). Press the 'List' push button to start the graphical Log Browser. (see section "Power-up Status")
Errors	Number of error-type log messages (level 0-3). Press the 'List' push button to start the graphical Log Browser. (see section "Log Browser")

Dumps	Number of core dumps. Press the 'List' push button to list the existing dumps. (see section "Dumps")
Number of hours back	Using the 'Number of hours back' slider you can set the period for the display of errors, warnings and dumps. If the slider is on 0, no time limitation is set, i.e. all information is shown, no matter how old. The default setting is the number of hours since the last system start.

Table 24: Elements of the window 'System Information'

#### 8.4.2.1 Power-up Status

If power-up failures occurred, they can be displayed in the list window:

Cache test failed

Duart 2681 or 26c92 console test failed

Duart 2681 or 26c92 (Other than console) failed

Floating Point Accelerator test failed

DPRROM test failed

LAN coprocessor failed

Memory test failed

NVRAM or battery test failed

SCSI controller 0 test failed

SCSI controller 1 test failed

Onboard EISA 8254 timer test failed

Time of day clock test failed

Onboard ISA ICU test failed

EISA backplane ICU test failed

Onboard ISA 8254 timer test failed

EISA backplane 8254 timer test failed

Centronics data register test failed

Onboard EISA ICU test failed

Onboard 8254 timer test failed

EISA backplane Scratch register test failed  
unknown

#### 8.4.2.2 Sar History

The show box for sar history allows you to view the automatic measurements of system

activities stored in the system. Each measurement indicates the day and status of any performance problems.

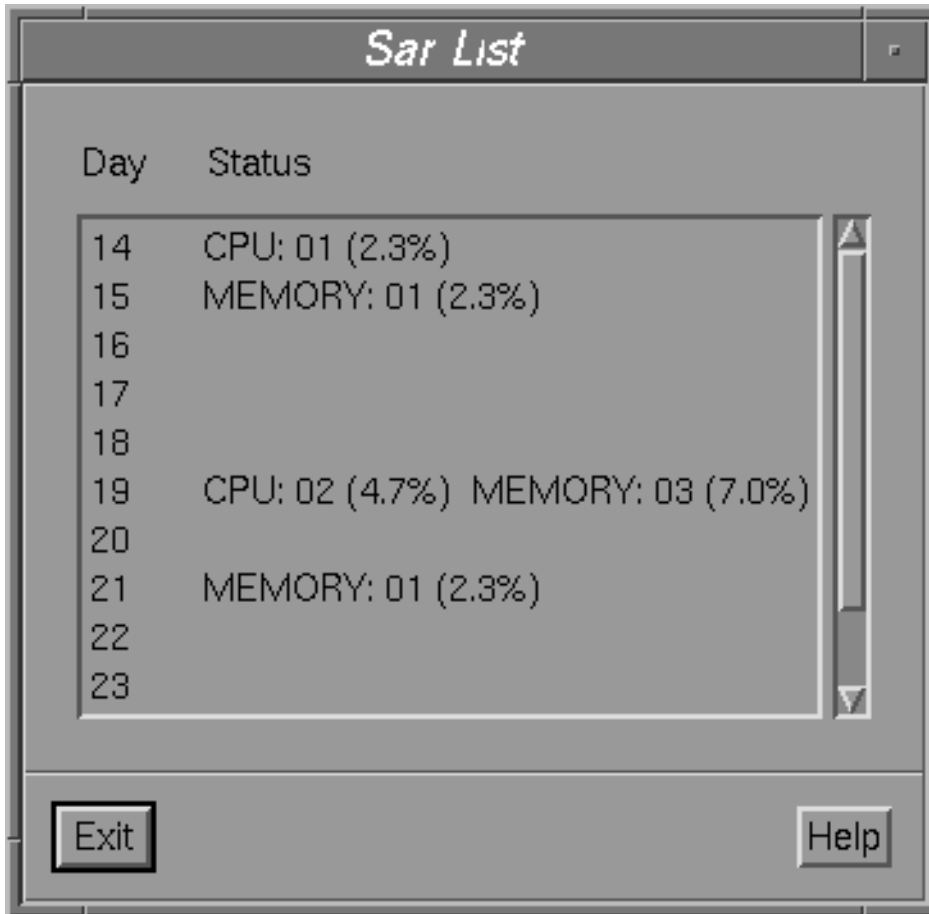


Figure 14: Sar History

The status display comprises the following information:

Day	Day of record	
Status	<Components> <Number> [<Percent>] Meaning:	
	Components	Processor or MEMORY
	Number	Number of measurements indicating possible performance problems.
	Percent	Percentage of problematic measurements among all measurements on the day in question.

Table 26: Components of the status display

### 8.4.2.3 Detail Show Box for Sar History

The detail show box for sar history allows you to view the automatic measurements of system activities stored in the system for a single day. The box shows important measurement parameters for assessing problems affecting CPU and memory:

%wio	Percentage of CPU time while waiting for block input/output.
%idle	Percentage of CPU time while idle. The CPU is waiting for processes to become executable.
int/s	Number of interrupts per second
runq-s z	Average number of executable processes in memory in run queue waiting for free CPU time.
ppgin/ s	Number of pages swapped in per second.
vflt/s	Page faults when converting a virtual address to a physical address.
ppgou t/s	Number of page-out requests per second

Table 27: measurement parameters

#### 8.4.2.4 Dumps

The detail show box for system dumps enables you to display detailed information on system dumps.

The following display options are available:

Info	General system information
Console Messages	2000 characters of stored console messages
Process Table	The process table at the time of the dump
System Variables	System variables that can be optimized

Table 28: Dumps

#### 8.4.3 Log Browser

The desktool 'Log Browser' enables you to view messages that are sent by applications or the operating system to the logging system.

You can select the messages for display either on the command line of the Log Browser or interactively in the selection box.

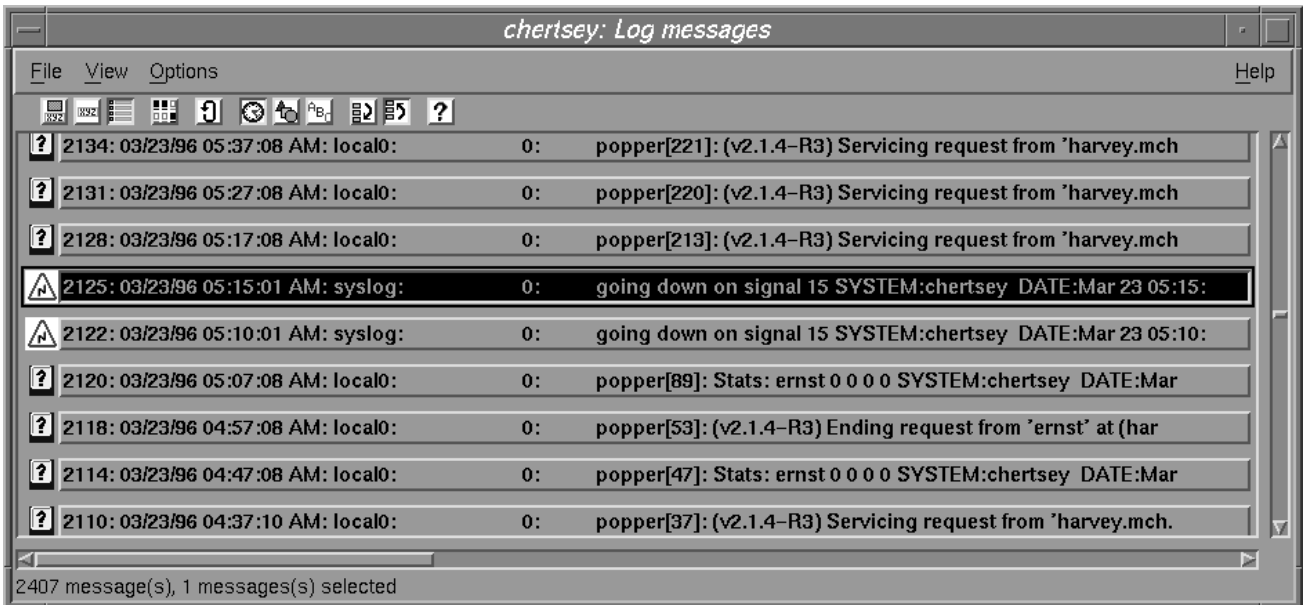


Figure 15: Selection box of the Log Browser

The selected messages are shown as icons in the main window of the Log Browser. An icon is assigned to each message level:










	Emergency
	Alert
	Critical
	Error
	Warning
	Notice
	Info
	Debug
	Unknown

Table 29: Messages

The icon labels show the application name and the error count of the message. In the extended display the following information is shown for each message :

Icon	Error	Date	Applic	Erro	Text. Shows the first
------	-------	------	--------	------	-----------------------

for message level	Count	Time	ation	r no.	60 characters of the log entry
-------------------	-------	------	-------	-------	--------------------------------

Double-clicking on a message displays the message data in a message box. In the message box 'Log Browser' shows the current values for the attributes of a message as entered in the log files:

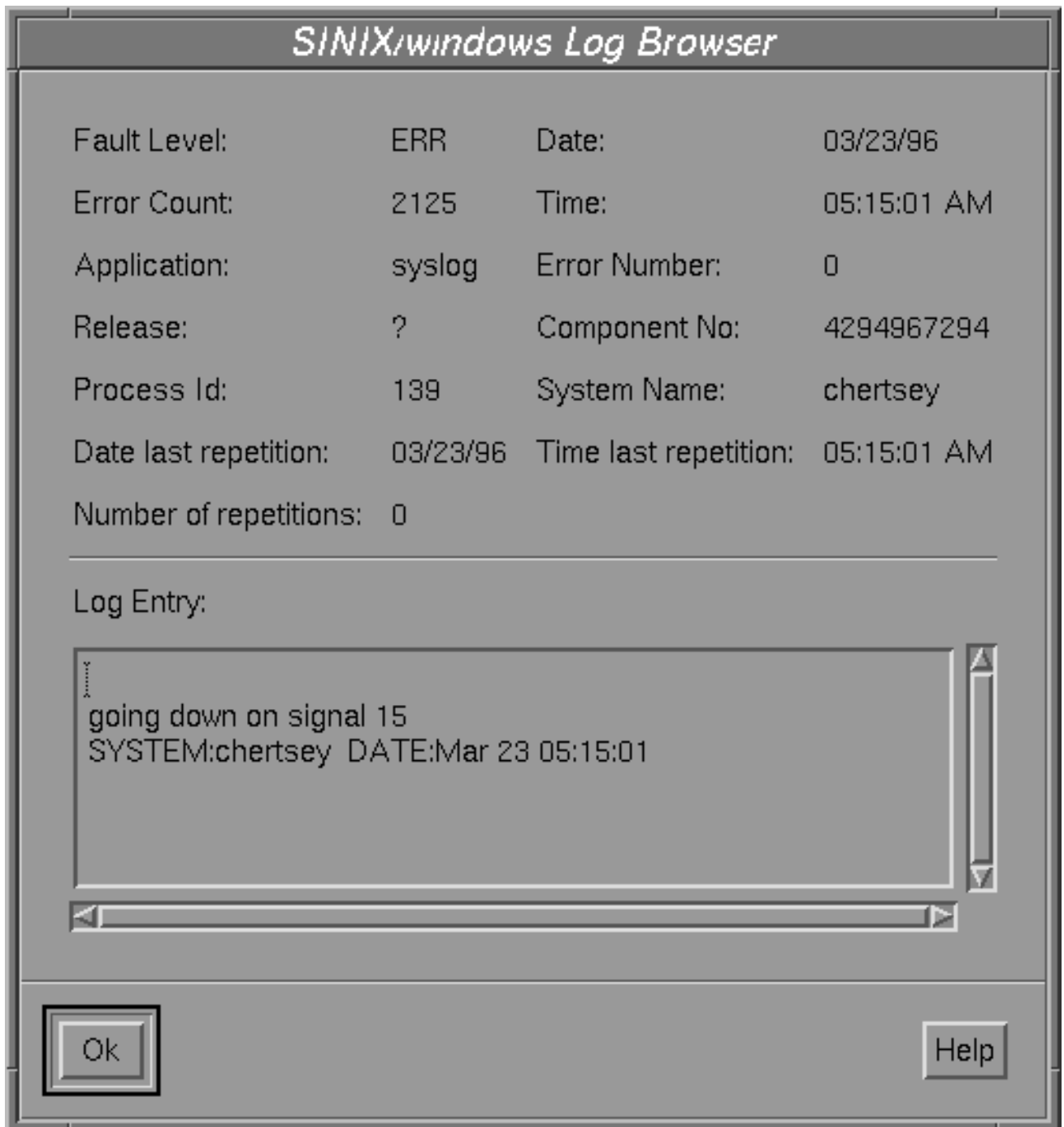


Figure 16: Message box 'Log Browser'

The following attributes are shown:

Fault level	Level of error message corresponding, in general, to the severity of the reported error. There are the following standard levels:
	EMERG system cannot be used
	ALERT immediate action required
	CRIT critical condition
	ERR error condition
	WARNING warning condition
	NOTICE normal but important condition
	INFO for information
	DEBUG debug messages
	TLOG20 messages from TLOG20 (older logging version)
	SYSLOG messages from syslog daemon
	Other levels are shown as a number
Date	Date of the entry
Error Count	System-assigned consecutive number
Time	Time of the entry
Application	Name of the reporting application or module
Error number	Error number assigned by the application
Release	Release/version of the reporting application or module
Component No.	Centrally-assigned number that uniquely identifies each logging application. The range of values extends from 1 to 4294967295 but the values 4294967294 (syslogd applications) and 4294967295 are reserved.
Process Id	Id of the process that sent the message.
System Name	Name of the system
Date last repetition	Date of last repetition of the message. If the application is recorded in the log3admin

	administration file for repetition detection, the last entry is compared with the current entry each time a write is made to this file. If they are the same, only the repetition counter and the repetition date or time are updated in the last log entry.
Time last repetition	Time of last repetition
Number of repetitions	Repetition counter
Log Entry	Message text. If information is stored for this message in the central error description file, it is also displayed. This additional display shows more detailed information on the problem and the actions taken.

Table 31: Elements of the box 'Log Browser'

#### 8.4.3.1 Select Log Messages

A selection box for log messages for display is a dialog box that shows the current criteria for selecting messages. It also allows the selection criteria to be changed.

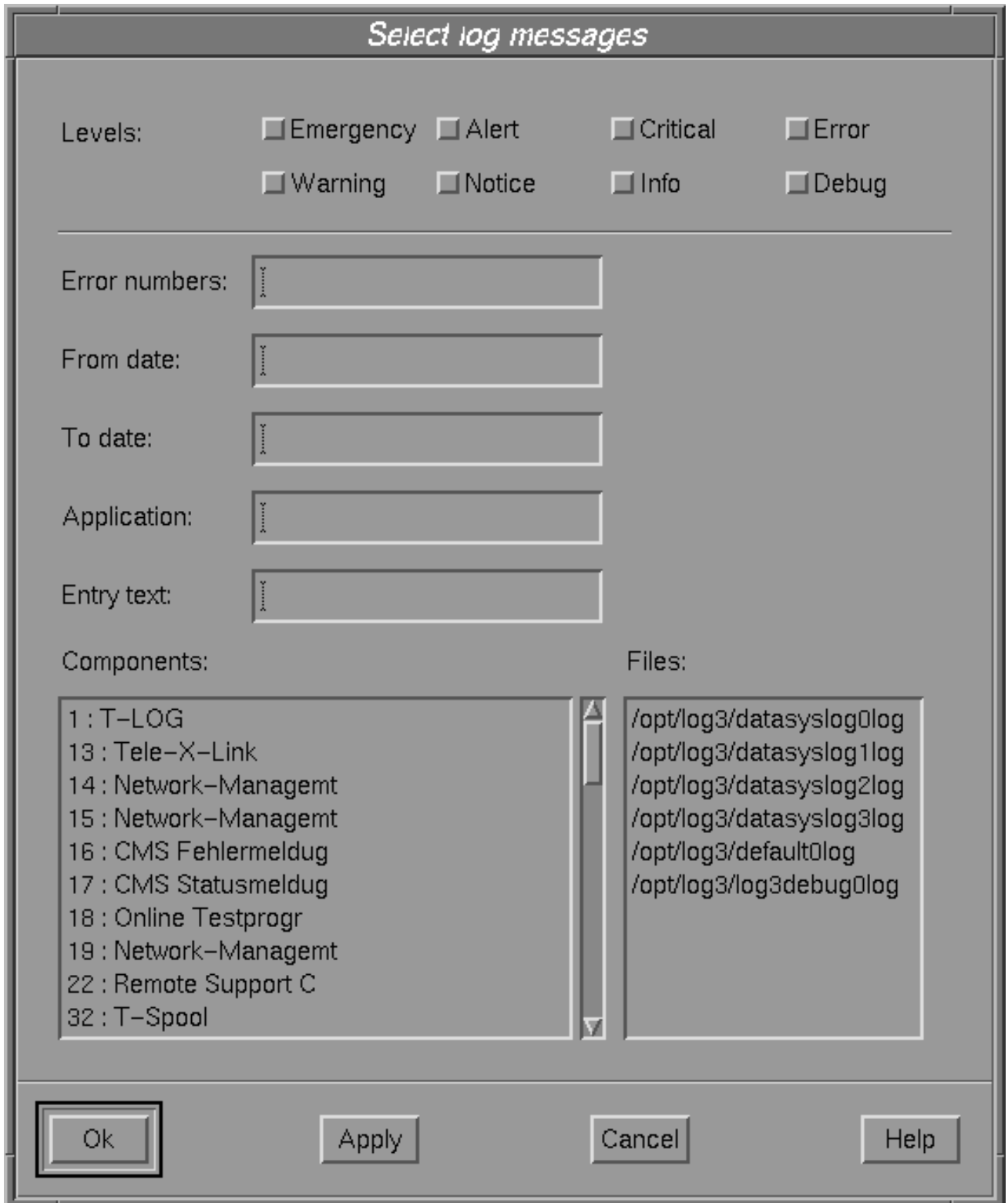
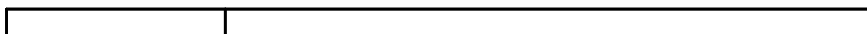


Figure 17: Selection box for log messages

The following criteria are displayed:



<b>Fault Level</b>	Level of error message corresponding, in general, to the severity of the reported error. There are the following standard levels	
	0	Emergency (system cannot be used)
	1	Alert (immediate action required)
	2	Critical (critical condition)
	3	Error (error condition)
	4	Warning (warning condition)
	5	Notice (normal but important condition)
	6	Info (for information)
	7	Debug (debug messages)
	If no level is selected, all are displayed.	
<b>Error numbers</b>	Error number assigned by the application. Entry can be made as comma-separated numbers or as ranges. The error numbers are between 0 and 65535.	
<b>From date</b>	Start date from which messages are to be displayed. Entry: yy[mmddHHMMSS] Missing entries are supplemented where possible.	
<b>To date</b>	End date up to which messages are to be displayed	
<b>Module</b>	Name of the reporting application or module	
<b>Entry text</b>	Search string in the text of the log entry. The search string can be specified as a regular expression. The structure of regular expressions is described for the system commands <code>ed</code> and <code>regex</code> . The simplest form is a normal text, but special characters like <code>*</code> , <code>-</code> , <code>'</code> and parentheses may not be used as they are interpreted.	
<b>Components</b>	All component numbers defined in <code>log3admin</code> are offered in a selection list and can be selected.	
<b>Files</b>	All log files are displayed and can be selected.	

Table 32: Criteria of log messages

## 8.5 System configuration

Desktools involveld:

National	xlc client (Local)
Date and Time	xdttm client
Font Service	fsconf client

Table 33: System configuration desktools

These desktools enable you to perform the following tasks:

- modify the country-specific user environment
- set the date and time
- install fonts
- manage the font server

Instructions on how to perform the individual tasks are given below.

### 8.5.1 Modifying the country-specific user environment

The "National" desktool enables you to customize the user interface to your country-specific requirements. If you are logged in as the system administrator, you can set the environment variables graphically-interactively on a system-wide as well as a user-specific basis. If you do not have system administrator privileges, you will be prompted to enter the system administrator password.

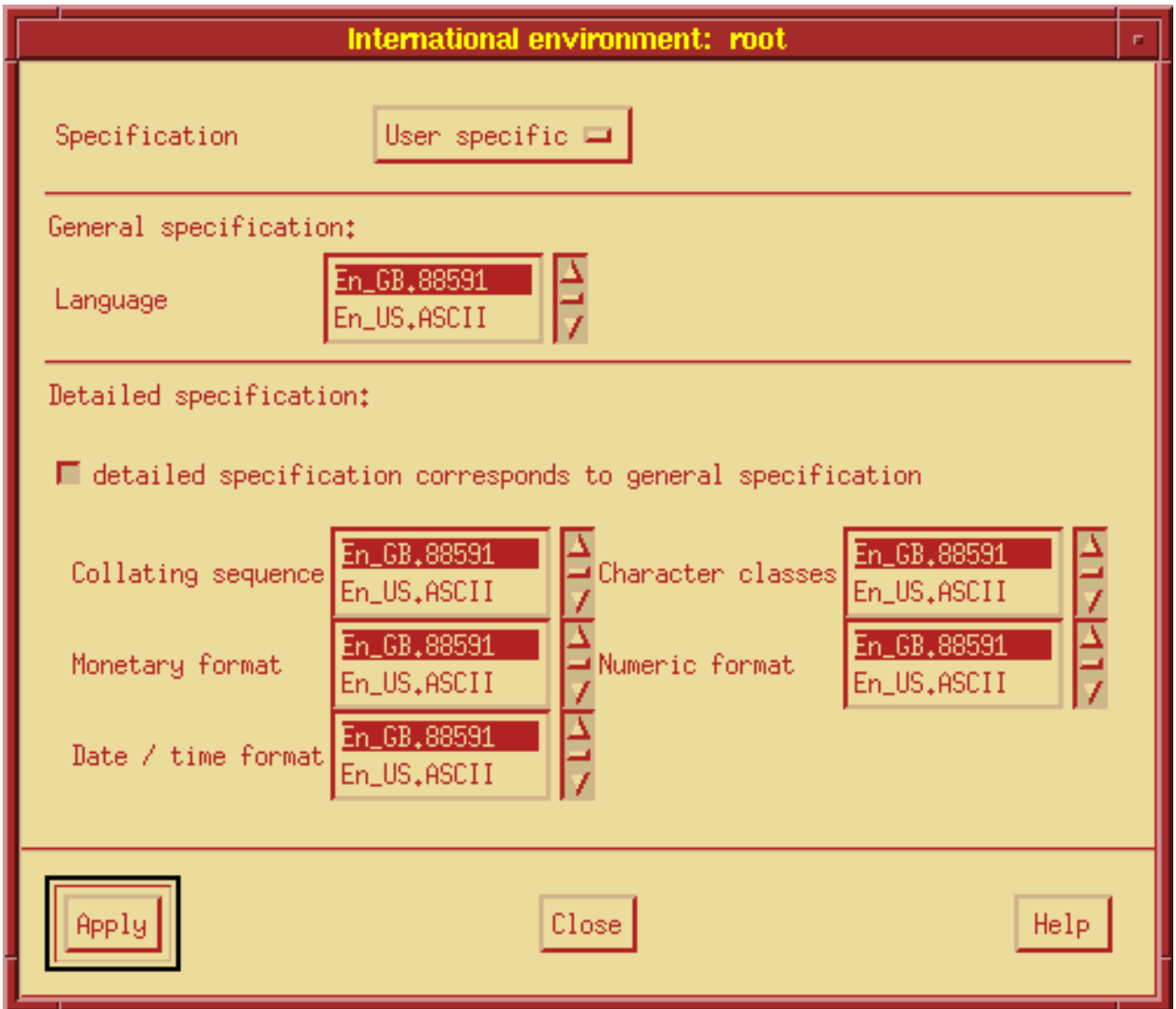


Figure 18: "National" desktool

To change the country-specific user environment, first select a general setting. Then use this setting to manipulate the values of the LANG environment variable. At the same time you set the LC\_MESSAGES environment variable, which relates to message texts. The general setting is the default setting for the more specific formats. The table below lists the individual environment variables which you can change selectively with this desktool, along with their functions:

LC_COLLATE	Sorting order
LC_CTYPE	Character class
LC_MONETARY	Currency format
LC_NUMERIC	Numerical format
LC_TIME	Date and time format

Table 34: Environment variables for modifying the country-specific user environment

The available language variants appear in the menus associated with the various option buttons.

You will find additional details in the help texts and in the chapter entitled "National environment - NLS (Native Language System)" in the "Programmer's Guide: Internationalization - Localization".

The user manager also uses the "National" desktool dialog box when creating users and user templates.

### 8.5.2 Setting the date and time

If you are logged in as the system administrator, you can use the "Date & Time" desktool to change the date and time settings used on your local host. For example, the "Clock" desktool uses these settings. Specific actions include setting the date and time and selecting a time zone. If you do not have system administrator privileges, you will be prompted to enter a password. You cannot make any changes unless you know this password.



Figure 19: Date and time

You will find detailed information in the help texts and under the heading "date" in the SINIX command manual.

### 8.5.3 Managing fonts

The "Font Service" desktool is an interface for installing fonts and for configuring and

managing a font server. Details can be found in section "Fonts".

### 8.5.3.1 Installing fonts

You can install only those fonts which are available in one of the following formats:

Format	Description	File suffix
BDF (Bitmap Distribution Format)	Bitmap source format	.bdf
PCF (Portable Compiled Format)	Portable hardware-dependent bitmap format	.pcf
SNF (Server Natural Format)	Hardware-dependent bitmap format	.snf
Speedo	Outline format from Bitstream Inc.	.spd
Type1	PostScript type 1 format	.pfa or .pfd

Table 35: File formats supported for fonts

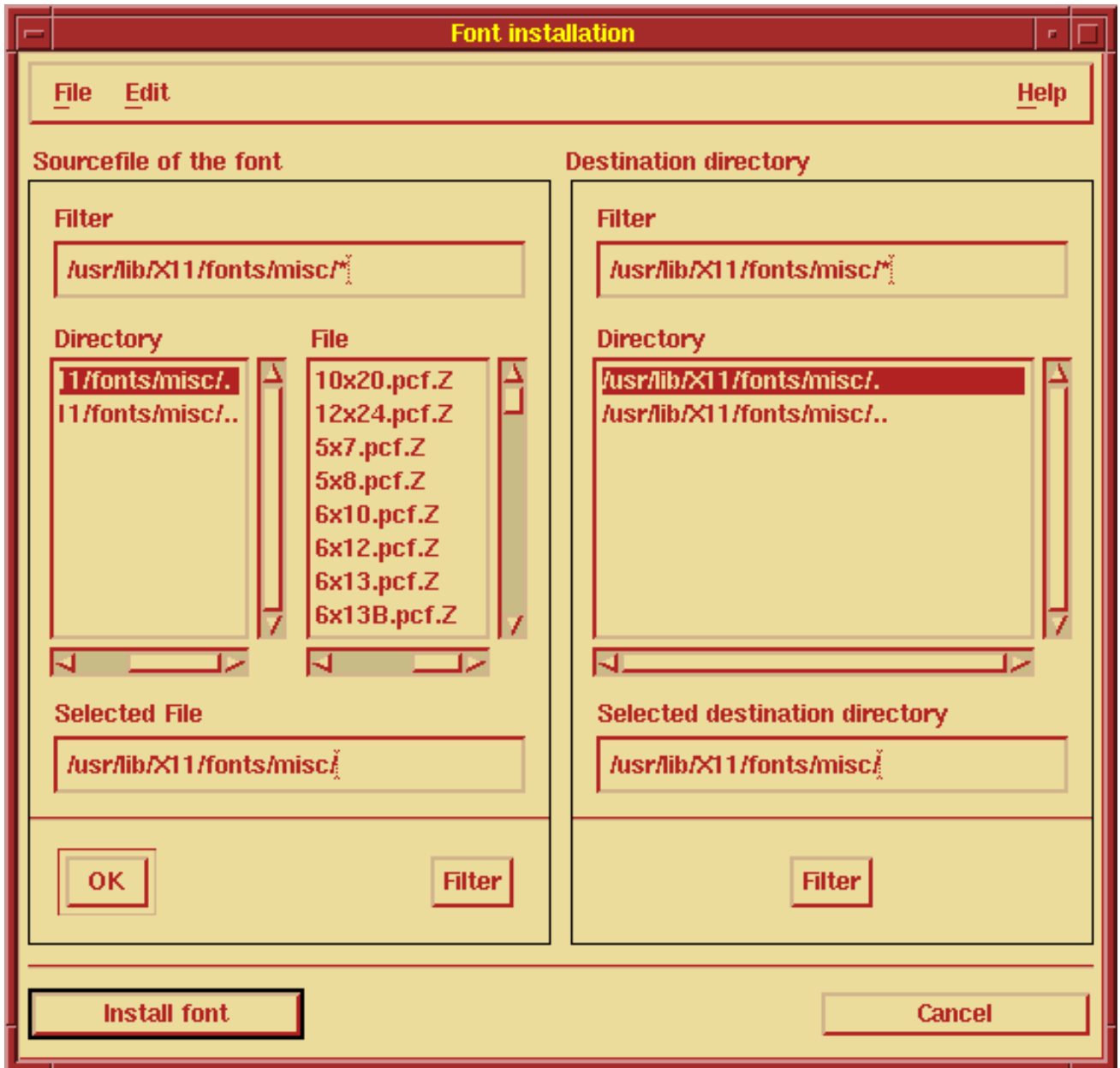


Figure 20: Installing fonts

You can use the file selection dialog box "Source file of the font" to select the font you want to install. In the "Filter" text input field, edit the source directory and the search pattern (e.g. /home/userid/fonts/\*). Click on "Filters" to select all files matching this pattern which are included in the "File" file list.

In the "File" file list, click on the desired file to select it. The selected file is copied to the "Selected File" text input field.

In the "Destination directory" file selection dialog box, select the target directory. You can edit both the target directory and the search pattern in the "Filter" text input field (e.g. /usr/lib/X11/fonts/\*). The current target directory is displayed in the "Selected destination directory" text input field.

**Note:**

Before you start one of the "Install font" and "Font alias" actions, you must define both the source file and the target directory.

Select the "Font alias..." button in the "Edit" menu to assign aliases to fonts. Select "Close" from the "File menu" or click on "Cancel" to terminate the application. You will find more information in the help files.

### 8.5.3.2 Configuring the font server

The Font Server administration tool is a graphical interface for configuring and administering a font server. This application will assist you in creating the configuration file. This file is evaluated when you start the font server. You can administer only the font server belonging to the configuration file currently loaded. The "Aktueller Font-Server" field displays the host name and the TCP port number of the current font server.

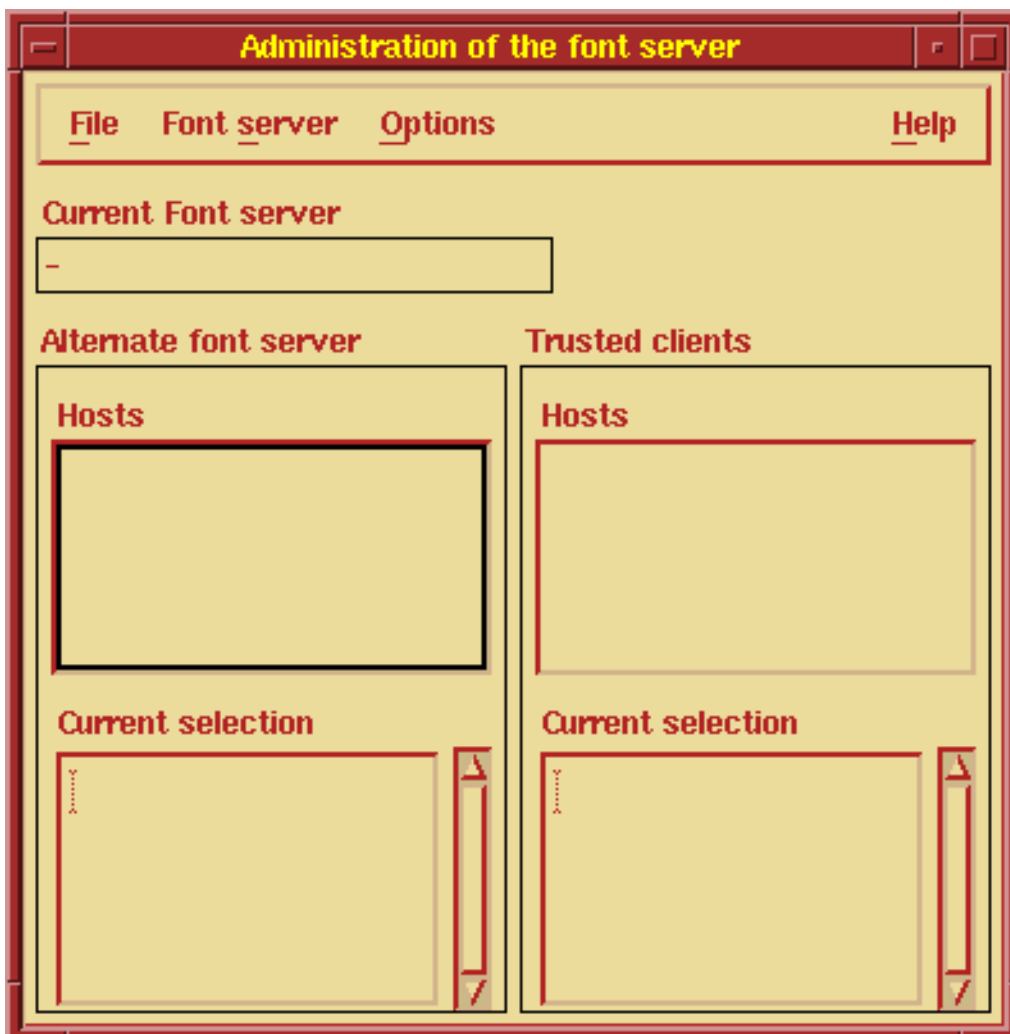


Figure 21: Font Server

The following parameters can be configured

:

---

Alternative font servers	Can be configured in the main window
Approved connections	Can be configured in the main window
Font catalog	Can be configured from the Options menu
TCP port	Can be configured from the Options menu
Client maximum	Can be configured from the Options menu
Default size	Can be configured from the Options menu
Default resolutions	Can be configured from the Options menu
Additional font server	Can be configured from the Options menu
Error log file	Can be configured from the Options menu
System message mechanism	Can be configured from the Options menu

Table 36: Configuring the font server

You will find details in the help texts.

## 8.6 Managing removable volumes

Desktool involved

Volume Manager	xdvm client
----------------	-------------

Table 37: Desktool for managing volumes and archives

### Note:

The client architecture is closely associated with the xdsm client. This means that the functionalities of the "Volume Manager" desktool are related to those of the "Software Manager". Functions relating to the management of software on external volumes (e.g. displaying, loading and installing software in the system) are described in section "Administering software".

The "Volume Manager" desktool enables you to perform the following tasks:

- manage devices
- manage archives
- manage software on removable volumes

Instructions on how to perform the individual tasks are given below.

### 8.6.1 Managing volumes on remote hosts

A description of how to use the Volume Manager to access other areas on the local host or a remote host can be found in section "Running managers in areas of the local host or a remote host".

### 8.6.2 Managing devices

The volume manager enables you to display all devices on the host from which the desktool was started. The existing devices are entered in the `/etc/device.tab` file. The possible devices are:

- CD-ROM drive
- floppy disk drive
- streamer drive

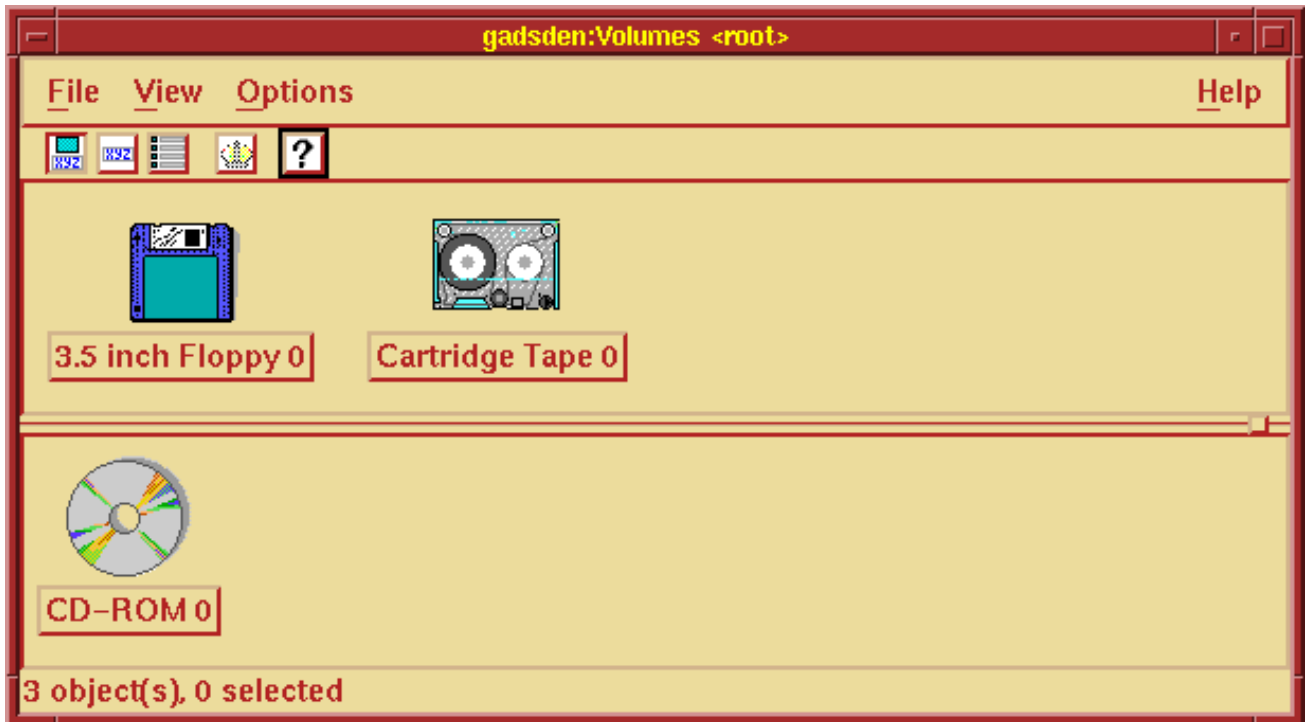


Figure 22: Volume manager

#### 8.6.2.1 Creating and mounting a file system

You can process archives and file systems on these volumes. If the volume does not yet contain a file system, you must first create one.

To create a file system on a volume:

1. Select the "Operations" cascade menu from the pop-up menu associated with the object.
2. Select "Make file system on volume..." from the cascade menu.
3. In the dialog box that now appears, select the "Apply" pushbutton. A file system is now created on the volume.

Next you must mount the file system before you can use it. You can mount a file system only if the device manager is running in system manager mode.

To mount a file system on a volume in the local file system:

1. Select "Mount" from the pop-up menu associated with the device containing the volume with the file system. If there are several options for selection when you mount the file system, "Mount" is a cascade menu.
2. From the cascade menu, select the desired option; the available options depend on the device type.

As archives and file systems require different operations, the device manager window is split in two when devices for both archives and files systems are available.

#### 8.6.2.2 Archives

This window pane shows devices which you can use to process archives on the volumes installed in this location. For example, you can process streamer cartridges or floppy disks here. Select items in the pop-up menu associated with the icons to open archive manager windows, where you can process the selected archive. You can access the contents of an archive only sequentially, but you can access the contents of a file system directly. This

means that you can address a file directly, for example. You can process only one archive per volume at a time.

### 8.6.2.3 File systems

This window pane shows devices which you can use to process file systems on the volumes installed in this location. For example, you can process CD-ROMs or floppy disks. Select items in the pop-up menu associated with the icons to open a file manager window, where you can process the selected file system. You can access the contents of a file system directly, but you can access the contents of an archive only sequentially. You can unmount a file system again when applications no longer access it.

### 8.6.3 Managing archives

The archive manager enables you to display the contents of archives, load archives or write archives.

In an archive manager window, icons are used to represent the contents of archives, i.e. they appear as objects for "Directories" and "Files". Since you can access the contents of an archive only sequentially and not directly, the directories and files are displayed differently than they are in the file manager.

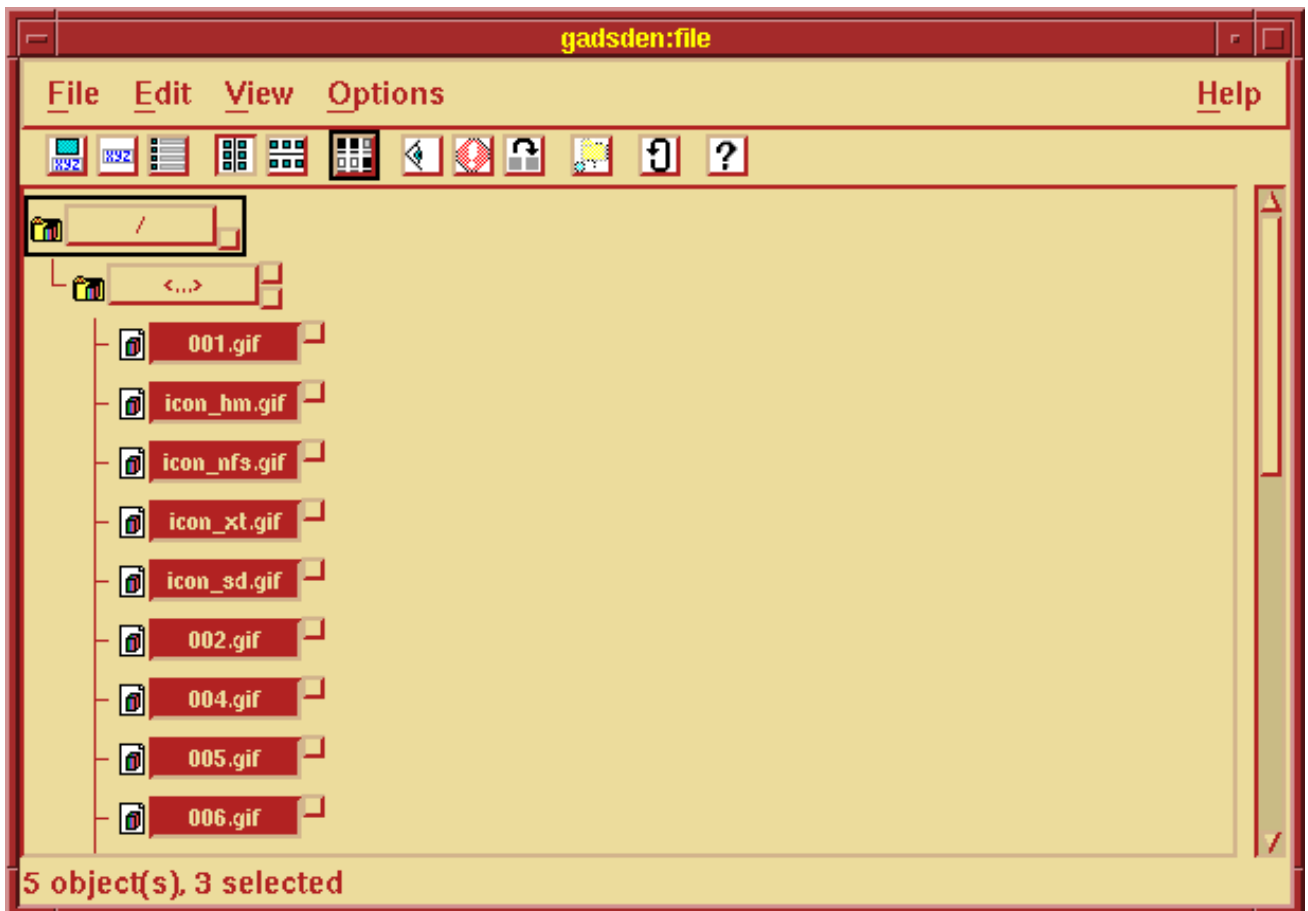


Figure 23: Archives

The archive manager typically opens a window for processing a previously selected archive.

However, you can also select a different archive for processing in this window.

To load files or directories from an archive:

1. Select the object you want to load.
2. Select "Read in..." from the "Edit" pull-down menu. The "Read Archive" dialog box is displayed.

To create an archive:

1. Select "Write archive..." from the pop-up menu of the device containing the volume to which you want to write the archive.
2. Enter the required values in the dialog box that now appears.
3. Click on the "Apply" pushbutton. The archive manager now writes the selected components to the archive and displays any components already written in the list box above the pushbuttons.
4. If you want to cancel writing the archive, select the "Stop" pushbutton.

**Warning:**

You cannot resume writing the archive from the point where it was cancelled.

**8.6.4 Managing software on removable volumes**

You will find the item "Show Software..." in the pop-up menu associated with the volume's icon. Select this entry to start the software manager for this volume. All related functions are described in section "Administering software".

## 8.7 Administering software

Desktool involved

Software Manager	xdsm client
------------------	-------------

Table 38: Software Manager desktool

### Note:

The client architecture is closely associated with the xdvm client. This means that the functionalities of the "Software Manager" desktool are related to those of the "Volume Manager". This has a special effect on the way in which the software manager is called (see below).

The "Software Manager" desktool enables you to perform the following tasks:

- retrieve information on available software
- load software
- install software
- delete software

Detailed instructions on how to perform the individual tasks are given below.

### 8.7.1 Invoking the software manager

The way in which you invoke the software manager depends on whether the software concerned is in the local system or on a removable volume. You can also start the software manager on remote hosts.

#### 8.7.1.1 Software in the local system

To start the software manager, double-click on the desktop icon. The software manager opens its main window containing the software objects which the host can access within the system. The window title consists of <host\_name>:system:<selected\_software>.

#### 8.7.1.2 Software on removable volumes

If the software is located on a removable volume (such as CD-ROM, floppy disk or streamer cartridge), you start the software manager from the volume manager. The volume manager main window displays the icons of all removable volumes which the host is able to access. You will find the item "Show Software..." in the pop-up menu of the icon. Selecting this item starts the software manager, which opens its main window containing the software objects located on the volume. The window title consists of <host\_name>:<volume>:<selected\_software>.

#### 8.7.1.3 Software on remote hosts

A description of how to use the software manager to access other areas on the local host or a remote host can be found in section "Running managers in areas of the local host or a remote host".

## 8.7.2 Retrieving information on available software

After you have started the software manager (see above), the icons for the software objects available in their respective locations are displayed in the desktool main window. For example, if you started the software manager directly by selecting its icon, the window displays the entire software installed on the host. If you started the software manager from the volume manager, you will see all software objects on the relevant volume.

### 8.7.2.1 Changing the software location

You can change the respective locations while working with the software manager. To do this, select "Change Location..." from the "File" menu on the menu bar or select the appropriate button on the icon bar. A dialog box appears. The "Position" option button in the dialog box show the current location on its interface (e.g. system, CD-ROM). To change the location, active an item from the options menu associated with the button.

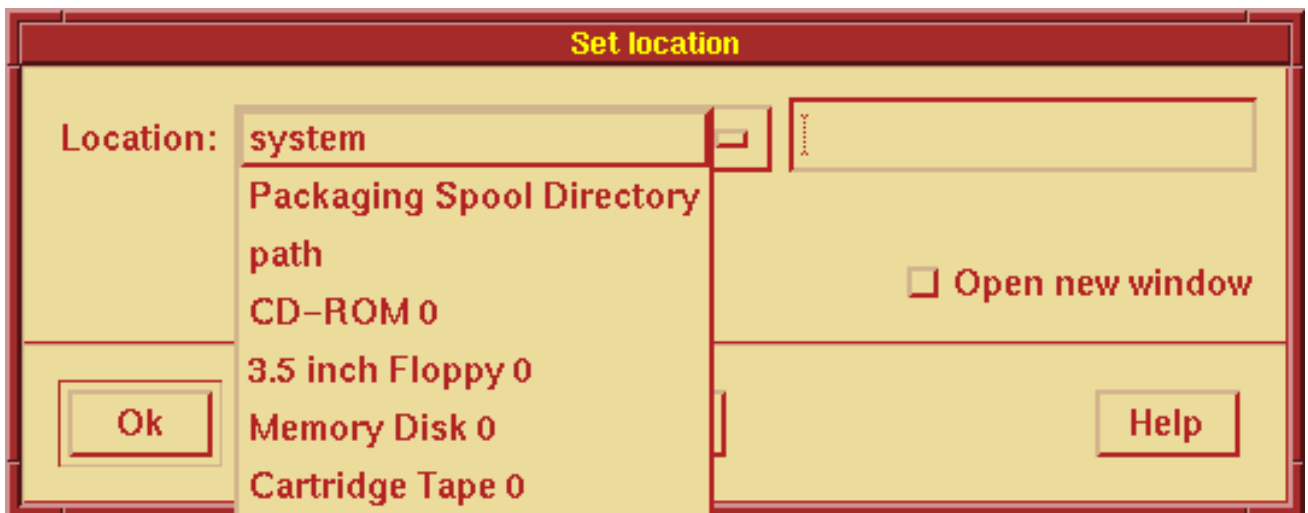


Figure 24: Changing the software location

From among the alias designations, select the possible software locations, e.g. the system, the default spooler directory (`/var/spool/pkg`) or a floppy disk drive. If you select "Pfad", you must specify a valid directory path in the field to the right of the button. It is also possible to access remote resources such as directories and CD-ROMs, provided that they are offered by the remote host and were mounted in the local system via NFS. A pushbutton enables you to decide whether to open another window for the new location.

### 8.7.2.2 Information on software objects

The icon format provides information on the status of the software it represents:

- The software is on the host or on an external volume.
- The software is a software package or a software product consisting of several software packages.
- The software is partially or completely installed in the system or not installed at all.

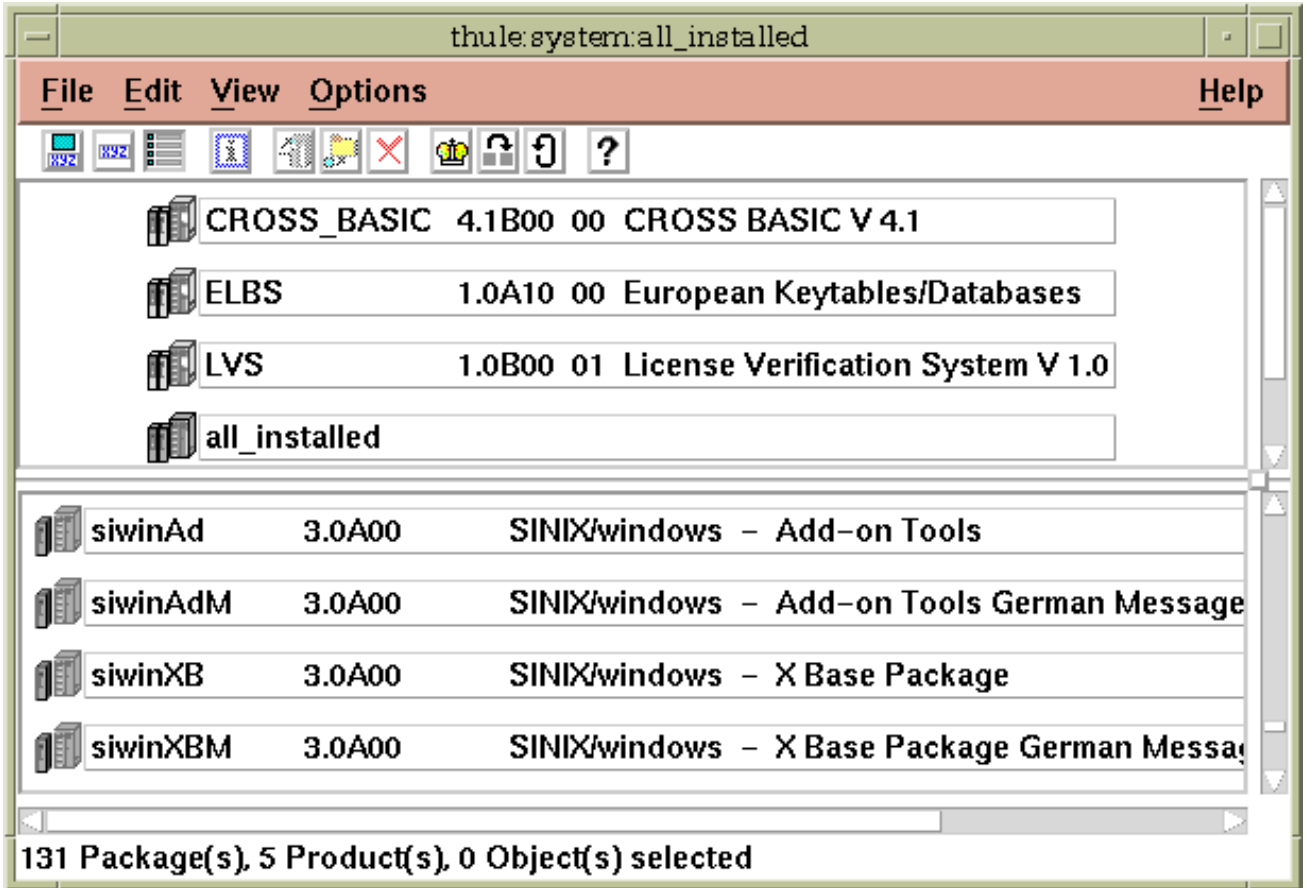


Figure 25: Software manager

Software products are displayed in the top portion of the window. Double-clicking on a product's icon displays the corresponding software packages in the form of icons in the lower portion of the window.

You can obtain detailed information about the individual software objects by selecting "Information..." from the pop-up menu associated with the icon or from the "Edit" menu on the menu bar. An information window displays information such as the software name and version, installation status and number of files and directories.

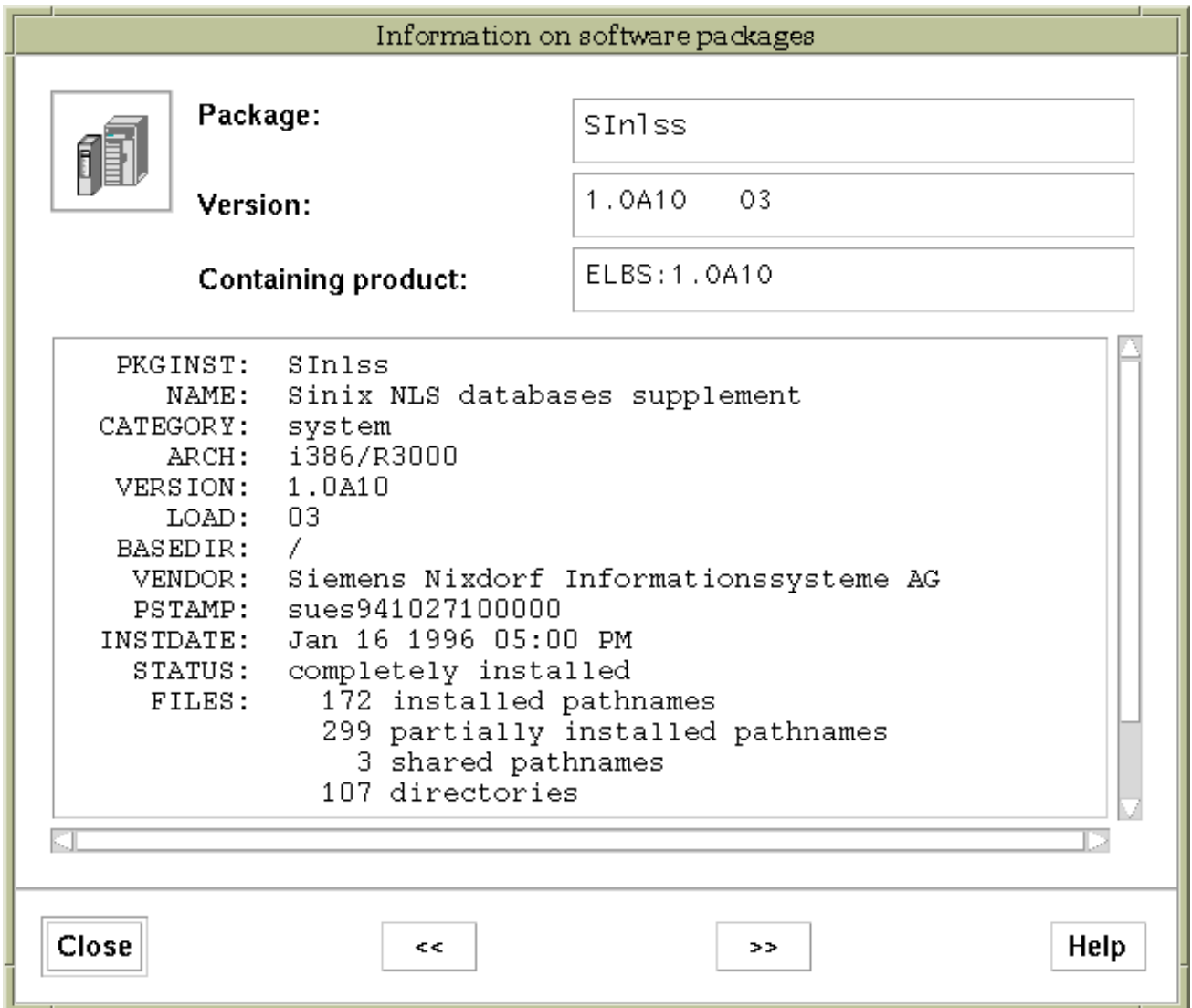


Figure 26: Information about software packages

To display the storage space requirements of all software objects installed in the system, select "Show disk space..." from the "File" menu on the menu bar. You can cancel the list by selecting the "Close" button at the bottom of the window.

### 8.7.2.3 Operations using software objects

Depending on their current location, you can load, install and delete software objects. A separate dialog box appears for each of the three operations. The description below on how to use the dialog boxes applies to all three operations.

There are three ways to open the dialog box from the software manager window:

- from the "Edit" menu on the menu bar
- from the pop-up menu associated with an icon, under the "Operations" cascade menu
- from the corresponding pushbutton on the icon bar

The title bar of the dialog box shows the current software location, i.e. the exact location from which the operation was started.

The previously selected software objects are displayed in the "Selected Software" list box at

the top of the window. You can add additional software objects to the list as described above. You can also drag and drop objects. To do this, drag an icon from the software manager window and drop it onto the list box. The name of the software object is placed in the list. Note the following: If objects are dependent on one another, they are placed in the list according to the order in which they were installed. If the object is a software package, its name replaces all previously selected packages belonging to this product. To remove software objects that were added to the list, select the object's name and click on the "Delete Entry" pushbutton.

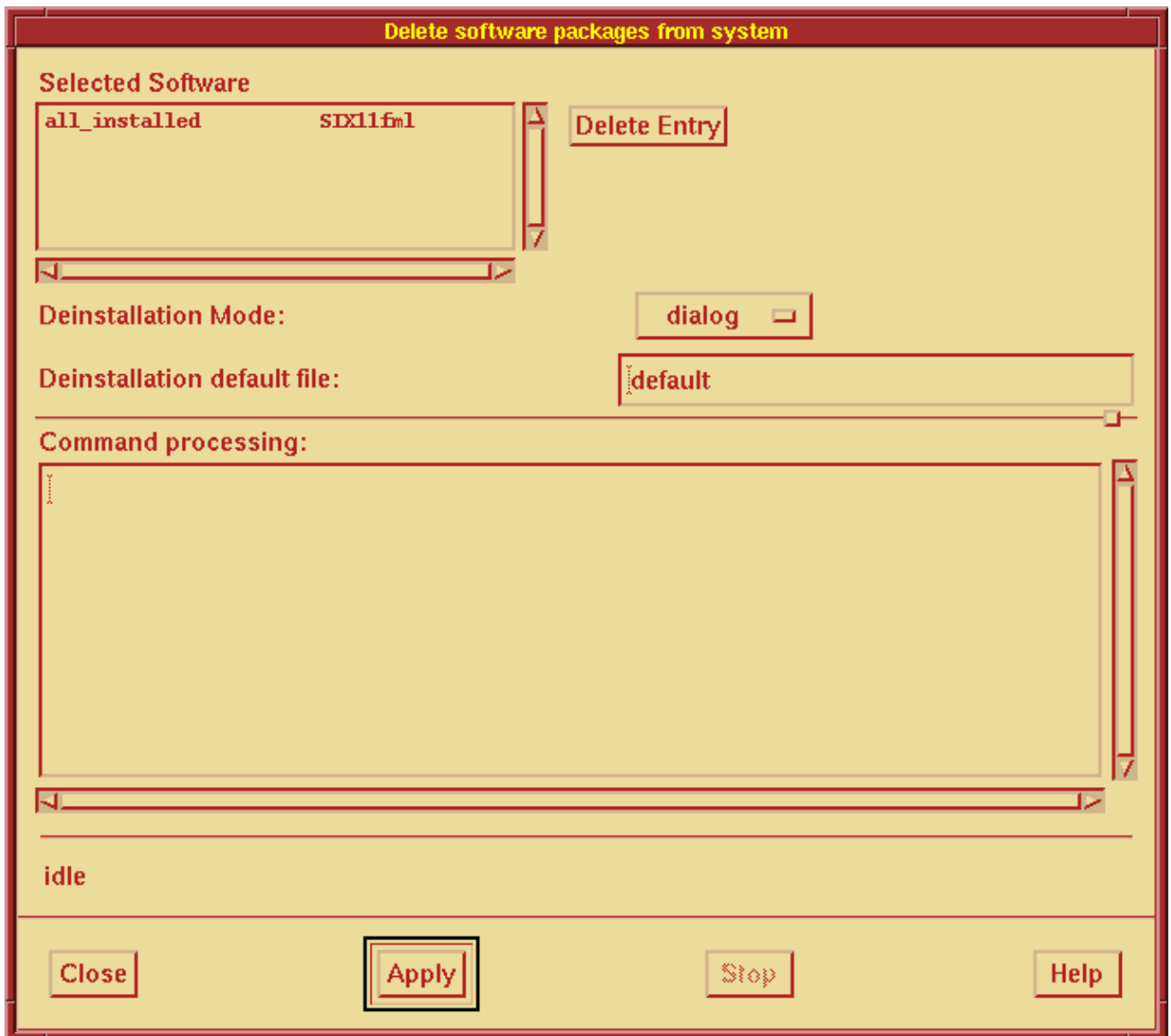


Figure 27: Example: deleting objects

The middle of the dialog box contains the parameter area. Its contents depend on the operation being performed.

Messages relating to the progress of the operation you have started appear at the bottom of the dialog box. The "Command processing" field displays the individual steps in the command sequence and any interactive user prompts to which you must respond,

depending on the default settings.

The "Command processing" field shows the current status of the action.

The pushbuttons that you can use to start, cancel and stop the operations as well as to request help are located at the very bottom of the dialog box.

### **8.7.3 Loading software**

Loading software means copying software packages from external volumes (such as floppy disks or streamer cartridges) to the directory on the host or between different directories.

The software manager supports software loading with the "Read in software..." dialog box.

When loading software, specify the directory where you want to store the loaded software in the parameter area. In addition to typing a valid path name you can also specify the alias for the spooler directory in the input field. The path's validity is checked.

To begin loading, click on the "Apply" pushbutton.

### **8.7.4 Installing software**

After you install software, this software is available to the user for completing his tasks. The system is therefore the target location for installation. In contrast, loading software merely involves copying the software to a specific directory.

The software manager supports software installation with the "Install software..." dialog box.

The layout of this dialog box is similar to that of the box used for loading software. To

familiarize yourself with the procedures for opening the dialog box and selecting the software to be installed as well as its general layout, refer to section "Operations using software objects".

The dialog box relates to the software location from which you opened the dialog box. To avoid errors which can arise when starting several installation procedures at the same time, it is possible to open only one "Install software..." dialog box at a time.

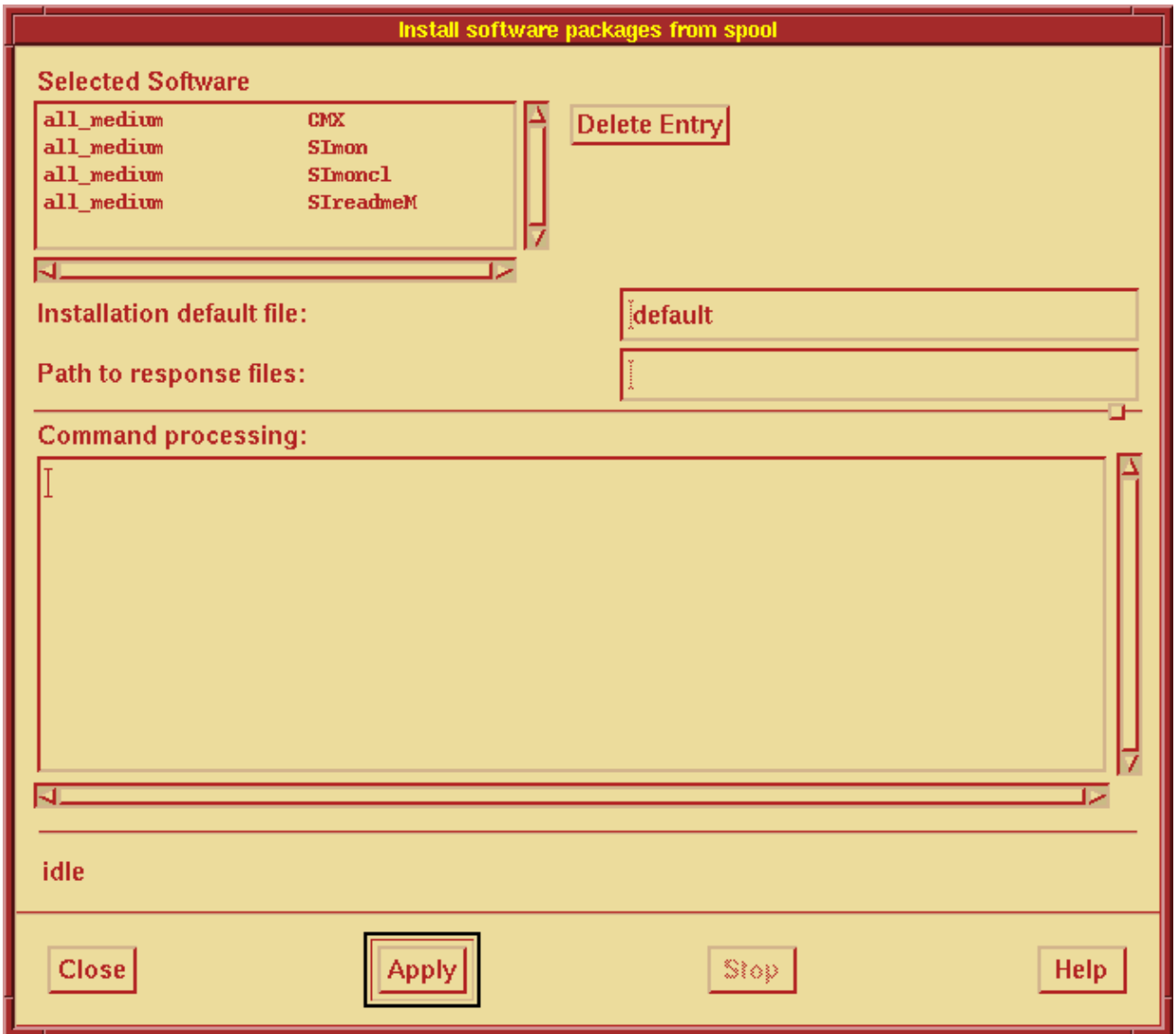


Figure 28: Installing software

When installing software, you must pay attention to three parameters which you can manipulate with controls in the parameter area of the dialog box:

Installation mode	The "Installation mode" option button enables you to decide whether to have the installation procedure run automatically or interactively.
Installation default file	The "Installation default file" field enables you to determine the default settings to be used for installation, i.e. the file in which these default settings are stored. The default file is the file named <code>default</code> in the <code>/var/sadm/install/admin</code> directory.

Path to response files	The contents of the "Path to response files" field are important if you selected "Automatic" as the installation mode. According to this procedure, the system simulates the user prompts, which are otherwise interactive, by automatically reading the response to dialog questions. When you install the software, these responses are generally stored in files which most products provide, for example, on the CD-ROM from which the software is installed. You must specify the path to the reply file; in the example this would be the <code>on_cd</code> alias.
Dialog	If you select "Dialog" installation mode, the responses that the user gives are stored in a file in the specified path and can be used later on in "Automatic" mode.

Table 39: Controls in the "Software installieren" dialog box

To start the installation procedure, click on the "Apply" pushbutton.

### 8.7.5 Deleting software

To delete software objects, the software manager provides a dialog box whose layout is similar to that of the box used for loading software. To familiarize yourself with the procedures for opening the dialog box and selecting the software to be installed as well as its general layout, refer to section "Operations using software objects".

Depending on the software location, the parameter area contains fields in which you must set parameters which are evaluated during deletion. The "Deinstallation Mode" and "Deinstallation default file" parameters are default settings. The two parameters function in much the same way as the installation parameters described in section "Installing software". After you define the software objects and parameters to be deleted, click on the "Apply" pushbutton to start the deletion process.

## 8.8 Printers

In the system as shipped, the "Printer" desktool is configured to access the SINIX system printer spooler on your local host.

The "Printer" desktool evaluates the `MOPRINTPATH` environment variable to find the directory containing the shell scripts that it accesses. If the `MOPRINTPATH` environment variable is not set, the desktool uses the shell scripts in the directory `/opt/bin/moprint` for the SINIX system printer spooler on your local host.

A directory `/opt/bin/moprint/printerSpooler` containing the shell scripts is created for the following printer spoolers:

<b>Product name</b>	<b><i>printerSpooler</i></b>
SINIX SPOOL V3.1	spool3.0
SPOOL V4.x	spool4.0
SPOOL V1.6	spool1.6
X/Open Spool (AT&T spooler)	spool5.4

Table 40: Printer spoolers

## 8.9 Troubleshooting

This chapter describes possible "emergencies" and gives you an idea of how to deal with them.

### 8.9.1 Error log files

Errors occurring while xdm is managing the display are recorded in either a system-wide or a user-specific error log file. The user-specific error log file holds all the error messages that are generated in the course of the user's current session. A copy of the user-specific error log file from the previous session is retained. The system-wide error log file holds all error messages generated since xdm startup which occur while xdm is managing the display but not during user sessions.

The name of the system-wide error log file is defined for xdm as the value of the errorLogFile resource of class ErrorLogFile. The user-specific file is the `.xerrors` file (the copy `.xerrors.old`) in the user's home directory.

### 8.9.2 System login

If it is not possible to log into a user session normally, it may be that the user-specific or system-wide session configuration file, xdm's Xsession file, has been modified. You can still log into the system by entering your login name and password, and then pressing [F1] instead of the return key. You will then obtain an emulator window and can, for instance, examine the error log files for possible causes. For more information refer to the subsections of this chapter headed "Error log files" and "Session management".

### 8.9.3 Session management

Abnormal events occurring in the course of a session may be caused by various factors. Some potential causes are summarized in the table below. The description is based on the progress of a typical session, as illustrated in the section "Session management". It is usually easier to pinpoint the cause of a problem if you can accurately indicate the stage the session had reached when it ceased to behave normally. Thus the first column of the table is labelled "Stage in session". The section "Interaction during configuration" will also be of some assistance in pinpointing why problems arise. "Action" typically indicates where to look in the documentation for more information which may help you in troubleshooting. References to clients relate to the corresponding client description in "SINIX/windows User Environment; Clients Reference Manual".

Stage in session	Possible cause	Action
Before logging in you do not get the typical root window background pattern.	The X server will not start	Refer to the manual for the X server you use for your display.

<p>You do not get an xdm host selection window</p>	<p>The X server makes no indirect request via XDMCP.</p>	<p>Check whether the X server that you use requests the desired xdm service. For more details, see the manual for the X server you are using for your display.</p>
	<p>xdm has a different entry in its Xaccess file.</p>	<p>Check whether the xdm on the host addressed by the X server provides the desired xdm service for the X server you are using in the form of an appropriate entry in xdm's Xaccess file. For further details see the xdm client description and the manual for the X server you are using for your display.</p>
<p>You do not get an xdm login window.</p>	<p>The X server cannot make a connection to xdm.</p>	<p>Check the communication link in the network between the hosts on which the X server and xdm are running. Refer to the manuals for your network.</p>
	<p>xdm cannot fulfill the X server's request for authorization.</p>	<p>Check whether the X server requests authorization and the xdm supports this. For further details, see the description of the xdm client and the manual for the X server you are using for your display.</p>
	<p>xdm is not authorized to</p>	<p>Check whether xdm supports the</p>

	access the X server.	authorization protocol of the X server.
	The X server is not authorized to access xdm.	Check whether the xdm on the host addressed by the X server provides the desired xdm service for the X server you are using in the form of an appropriate entry in xdm's Xaccess file. For further details see the xdm client description and the manual for the X server you are using for your display.
	The configuration for connection setup between the X server and xdm is incorrect.	Refer to the xdm client description.
You cannot log in with your usual login name and password.	The keymap has been modified and the special characters are not in the usual place.	Check the positions of the special characters on the keyboard. As the keymap has to be matched to your display, mapping is usually done by the X server. For more information on keymaps refer to the manual for the X server you use for your display, the xmodmap client description, and the init resource in the xdm client description.
	You want to log in as root	Check the file.

	and /etc/default/log in contains restrictions on the keyword CONSOLE.	
	The X server is connected to an instance of xdm on another host.	Check the display name of your X server, its network address and its symbolic name, and also the host name in the xdm login window. Refer to the manual for the X server you use for your display and to the xdm client description.
The xdm login window closes and xdm accepts your login name and password, but the xdm login window reappears after a short interval.	Your session is wrongly configured.	Log in as described in the section "System login". Using the error log files, for example, check whether the client configured not to run in the background has aborted prematurely. Typical examples of clients not run in the background are mwm, xdesk or a terminal emulator such as xpct. Refer to the appropriate client description, to the section "Points to note", and to the xdm client description.
You get the wrong locale (language environment).	The LANG environment variable has not been set.	Refer to documentation dealing with defining environment variables in

		configuration files.
Your shell startup script, e.g. .profile for the Bourne shell, is not executed.	In the graphical user interface the \$HOME/.xsession file takes charge of user-specific session initialization.	Refer to the chapter "Session configuration using the .xsession file".
	In the configuration as shipped with the product, the file \$HOME/.sw_profile takes charge of user-specific initialization of environment variables.	Refer to the section "".
You cannot start a client running for your display.	The PATH environment variable does not include the client path.	Check whether the PATH environment variable includes the value /usr/bin/X11, possibly between colons ":". Check this in the environment in which you call the client - like any command, the mwm client inherits the current environment on invocation, and this may affect its root menu, for example. Refer to the parts dealing with environment variables in your operating system commands manual.
	The client does not	Check whether the DISPLAY

	recognize the display name.	environment variable is set in the client's invocation environment. Refer to the description of the DISPLAY environment variable for the client.
	The client is not authorized to access the X server.	Check whether the host on which the client is running is in the X server's access control list, or whether the client can access the user-specific authorization information. Refer to the xhost and xdm client descriptions for details (keyword: authorization).
	The client has aborted prematurely.	If you invoke the client from your session configuration file, check for associated entries in the error log files. Refer to the section "Points to note".
When you start a client, a message tells you that a font cannot be found.	The X server does not know the font.	Check which fonts the X server knows. Refer to sections dealing with font paths or font servers in the manual for the X server you use for your display.
When you start a client, a message tells you that the colors cannot be set up as	The X server does not know the color or cannot display it owing to lack of space.	Using the xpct client, for example, check which colors the X server knows. Refer to sections dealing with the color database in

requested.		the manual for the X server you use for your display.
When you log out of the system, the X server configuration for the session remains in effect, e.g. the root window pattern of the keymap.	The X server is not being reset. That is how the X server is configured.	Refer to the manual for the X server you used for your display.
	That is not how xdm is configured.	Refer to the descriptions of the xdm resources reset (class:Reset) and terminateServer (class: TerminateServer).

Table 41: Troubleshooting

## 9 Related publications

- [1] SINIX/windows  
Documentation Overview
- [2] SINIX/windows User Environment  
Introduction to Handling and Configuration  
(SINIX Desktop)  
User Guide
- [3] SINIX/windows User Environment  
Clients Reference Manual  
(SINIX Desktop)
- [4] SINIX/windows User Environment  
XDCL Desktop Configuration Language  
(SINIX Desktop)  
Reference Manual
- [5] SINIX/windows User Environment  
Desktop Kornshell User's Guide  
(Common Desktop Environment)
- [6] SINIX/windows User Environment  
User's Guide  
(Common Desktop Environment)
- [7] SINIX/windows User Environment  
TED Enhancements  
(Common Desktop Environment)
- [8] SINIX/windows User Environment  
Advanced User and System Administrator Guide  
(Common Desktop Environment)
- [9] SINIX/windows Development  
Product Overview
- [10] SINIX/windows Development  
OSF/Motif  
Style Guide Release 1.2
- [11] SINIX/windows Development  
OSF/Motif  
Programmer's Reference, Release 1.2
- [12] SINIX/windows Development  
OSF/Motif  
Programmer's Guide, Release 1.2
- [13] SINIX/windows Development  
OSF/Motif  
X Server Reference Manual
- [14] SINIX/windows Development  
X Window System, Xt Toolkit Intrinsic  
Reference Manual
- [15] SINIX/windows Development  
X Window System X11, Xlib  
Programming Manual

- [16] SINIX/windows Development  
X Window System, Xlib  
Reference Manual
- [17] SINIX/windows Development  
X Window System X11 Rel.5 (SINIX), X Input Extension  
Programmer's Reference
- [18] SINIX/windows Development  
X Window System, X Protocol  
Reference Manual
- [19] SINIX/windows Development  
Widget Set  
Programmer's Reference
- [20] SINIX/windows Development  
Programmer's Guide  
(Common Desktop Environment)
- [21] SINIX/windows Development  
Programmer's Overview  
(Common Desktop Environment)
- [22] SINIX/windows Development  
Application Builder User's Guide  
(Common Desktop Environment)
- [23] SINIX/windows Development  
Help System Author's and Programmer's Guide  
(Common Desktop Environment)
- [24] SINIX/windows Development  
Internationalization Programmer's Guide  
(Common Desktop Environment)
- [25] SINIX/windows Development  
Style Guide and Certification Checklist  
(Common Desktop Environment)
- [26] SINIX/windows Development  
Tooltalk Messaging Overview  
(Common Desktop Environment)
- [27] SINIX/windows Development  
Tooltalk Programmer's Guide  
(Common Desktop Environment)

### **Ordering manuals**

The manuals listed above and the corresponding order numbers can be found in the Siemens Nixdorf *List of Publications*. New publications are described in the *Druckschriften Neuerscheinungen (New Publications)*.

You can arrange to have both of these sent to you regularly by having your name placed on the appropriate mailing list. Please apply to your local office, where you can also order the manuals.

# Index