iXOS-JUKEMAN 2.2

User Manual

Pre.12/97

iXOS Software AG • Technopark • Bretonischer Ring 12 • D-85630 Grasbrunn
Tel. +49 (89) 46005-0 • Fax +49 (89) 46005-199 • email: office@munich.ixos.de

ImpressumiXOS-JUKEMAN 2.2 User Manual
Pre.12/97

Contents

1 Introduction

1.1 About this manual

This manual covers the installation, set-up and usage of iXOS-JUKEMAN, a server software product for optical media. An introduction into the features and possibilities of the software can be found in the section “iXOS-JUKEMAN — an overview” on page 13.

1.1.1 Software release and further information

Covered in this manual is the release iXOS-JUKEMAN version 2.2. The latest release can always be found on our web server:

http://www.jukeman.com

Our web server contains all the latest information on iXOS-JUKEMAN pricing, resellers and technical information (Support, FAQs).

There are two newsgroups, *jukeman.sales* and *jukeman.tech* for sales information and technical information respectively. Our news server address is *news://www.jukeman.de*.

These newsgroups offer information additional to those found in the FAQs of this manual. It may be a good place to look for special solutions before you go and ask your distributor or our support.

New features in version 2.2

- Support for Joliet file system (read/write)
- WORM file system
- Support for multisession CDs (read)
- Support for new jukeboxes
- Enhanced support for OS/2 clients
- Support for Novell clients
- Support for Macintosh clients and HFS
- Hybrid CDs (ISO or HFS alternatively)
- Volume names can be changed with operating system functions
- Burning of multiple volumes from the graphical user interface
- Journal functionality (retains cache contents after a possible power failure)
- Windows NT properties for all files of the JUKEMAN directory (e. g. build version)

1.1.2 Structure of this manual

Overview of the software:

- “iXOS-JUKEMAN — an overview” on page 13.

  This section is recommended for first time users, who want to discover the possibilities and concepts of iXOS-JUKEMAN.

First time installation and set-up:

- “Installation” on page 21.
- “Setting up iXOS-JUKEMAN” on page 29.

These sections are required for first time installation and set-up of iXOS-JUKEMAN.

Using iXOS-JUKEMAN:

- “Using iXOS-JUKEMAN” on page 85

A task oriented section for using iXOS-JUKEMAN.

Further information:

- “Supported jukeboxes” on page 163.
- “Command line index” on page 213
- “Configuration file server.cfg” on page 237
- “Log file logfile.txt” on page 249.
- “FAQ/Troubleshooting” on page 253.

1.2 Conventions

The following conceptual and typographic conventions are used in this manual:

Conceptual conventions:

- **Disk**: iXOS-JUKEMAN supports all kinds of optical media: CDs, CD-Rs, PDs, MOs and WORMs which are generally referred to as disks in this manual.
- **GUI**: The graphical user interface provides an easy way to administer iXOS-JUKEMAN on Windows NT. Sections dealing specifically with this way of administration are headed “GUI”.
- **CLI**: The command line interface can be used on Windows NT and UNIX to administer iXOS-JUKEMAN. Sections dealing specifically with this way of administration are headed “CLI”.

Typographic conventions:

- `Commands and program names will be typeset in this typewriter font.`
- To represent a menu selection, the following format will be used: **[MENUNAME]-ENTRY**, e. g. **[DEVICES]-NEW** tells you to open the “devices” menu and select the entry “new”.
- Parameters to commands will be printed in angle brackets: <*parameter*>. Optional parts will be printed in square brackets: `[can be ommitted]`.

1.3 iXOS-JUKEMAN — an overview

iXOS-JUKEMAN is the leading software product for managing jukeboxes, drives, and recorders for optical disks. It provides simple and efficient access to optical storage devices with unrivaled performance and flexibility.

Simplicity

iXOS-JUKEMAN presents all available disks as a standard file system, completely hiding the jukeboxes. The UNIX version acts as a standard NFS file server; the NT version acts as a standard NT file system that can be shared across all available protocols. Users can access the disks as easily as they would access a shared hard disk. Version 2.2 also allows clients to concurrently write disks.

Efficiency

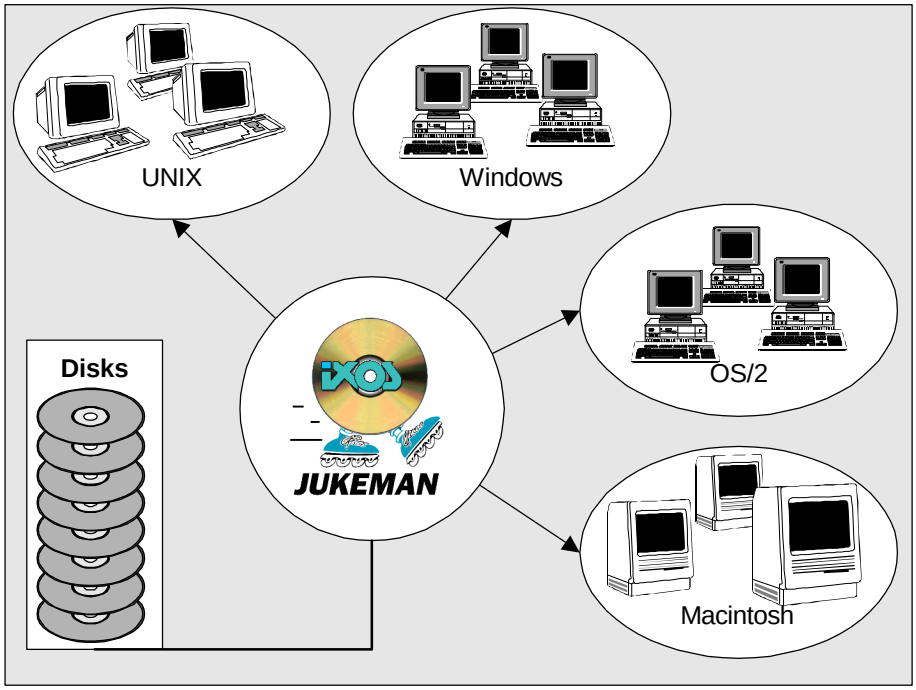
The file system server provides data and directory structure caching, and powerful device management and job scheduling features. As a result, iXOS-JUKEMAN provides the industry's highest performance - even under extremely high load use.

In addition, iXOS-JUKEMAN's redundant disk management means you can count on quick response times with the added advantage of protection against hardware failure.

Flexibility

iXOS-JUKEMAN has been carefully designed to ensure support for a wide range of environments, including a range of mixed clients and network protocols.

In addition, iXOS-JUKEMAN is highly customizable. For example, multiple views can be defined with various name formats and configurable subsets of visible disks.



1.3.1 Main components

iXOS-JUKEMAN consists of a file system **server** and a CD **writer**.

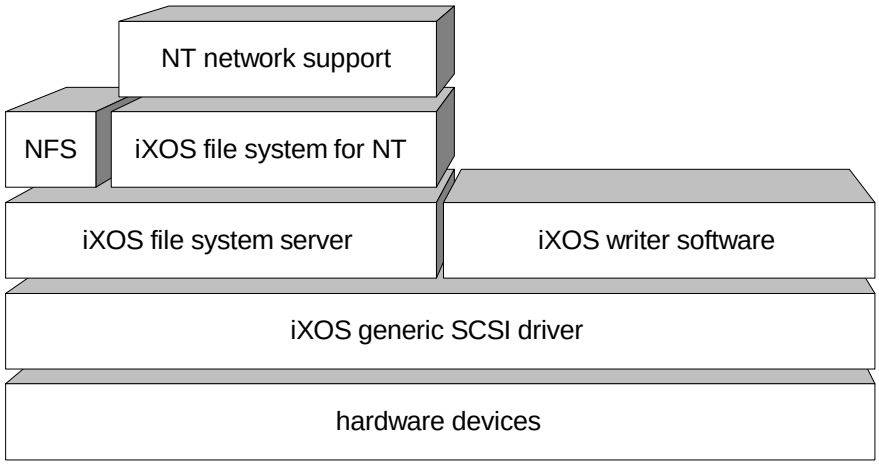
The **server** provides native file system support for multiple client types, and allows users to view CDs in their preferred file system format. It accesses real hardware devices and hides them completely by presenting an abstract hierarchical file system. It caches files and directories, optimizes access to CDs and jukeboxes, and minimizes jukebox movements.

The **writer** supports burning ISO 9660 file systems on recordable disks, assuring the reliable constant data rate required by CD recorders. In addition, the writer supports Rock Ridge attributes and the Joliet file format.

Version 2.2 of iXOS-JUKEMAN combines the server and writer in a **writable file system**. It can be configured to present all writable disks as if they were hard disks, allowing clients to add data to these disks through the same file system interface that presents the disks for read access.

For each operating system, iXOS-JUKEMAN uses one **generic SCSI driver** for all hardware-related software products. A device that works under one operating system will work equally reliably under all other supported operating systems.

The NT version also includes “Jukeman Administration”, a **graphical user interface** (GUI) that makes administration available, both locally or remotely from any PC.



The server

The iXOS-JUKEMAN server controls jukeboxes, disk drives, and the disks they contain. The server combines all disks in a single file system in which each disk is a subdirectory of the root directory and the jukeboxes are hidden.

NFS

UNIX clients access the server's file systems via NFS. The server accepts NFS requests and replies with the NFS protocol (version 2). Clients mount the file system just like any other Network File System.

Windows NT file system

With Windows NT, iXOS-JUKEMAN includes a native file system that appears as a drive letter, and can be accessed and shared through the file system file server. This guarantees that the same jukeboxes are supported across both the NFS and NT file system interfaces.

Because NT shares local native file systems across all installed protocols, PCs can access the file system as easily as they access a network drive. The same file system can simultaneously be accessed by UNIX clients via NFS. iXOS-JUKEMAN supports the MacFile service, allowing Macintosh clients to access the file system.

optimized access

IXOS JUKEMAN optimizes jukebox performance caching, optimizing disk movement, parallel access to all devices, advanced queuing, and "request anticipation". iXOS-JUKEMAN also recognizes hard disk images as disks – allowing them to be used as a replacement for CD drives. With the built in NTFS compression, single hard disks can replace large numbers of CDs.

For very high performance demands, disks can be replicated across several jukeboxes. When a disk is requested the server automatically chooses the least loaded jukebox.

The writer

The iXOS-JUKEMAN writer can send either ISO 9660 file systems or raw data to a disk recorder. Moreover, Rock Ridge attributes or the Joliet format can be used. The writer accepts data from a pre-mastered file, a raw partition, a disk drive, or from a pipe.

The writer's preview mode can be used to test the writing process before you start, and the verify option allows the CD to be verified immediately after it is written.

incremental file system

A special feature of iXOS-JUKEMAN is that the file system can also be used by clients to write to disks. Thus, iXOS-JUKEMAN supports an incremental file system for writable disks. To eliminate the high overhead of session lead-in and lead-out, iXOS-JUKEMAN provides multi-track writing which uses the CD space more efficiently than multi-session writing. Up to 96 of the 99 tracks of a CD-R can be written with data. For PDs, WORMs, and MOs, the number of tracks is not limited.

The generic SCSI driver

iXOS-JUKEMAN provides support for a wide range of devices including jukeboxes, stand-alone CD-ROM drives and hard disk drives across multiple operating systems.

To minimize the non-portable sections of the drivers, iXOS-JUKEMAN uses a generic SCSI driver that passes a SCSI command to the hardware through the operating system.

If a generic driver is not present, the iXOS generic driver must be installed. Once the driver is installed, it works with all iXOS products.

1.3.2 Supported hardware

iXOS is continually adding support for new jukeboxes and recorders. An up-to-date device list can be found on www.jukeman.com or from **support@europe.jukeman.com**.

The following tables list the jukeboxes and recorders tested with iXOS-JUKEMAN. In the "Type" column, a '?' represents any single letter, a '*' represents any string.

iXOS-JUKEMAN also supports ISO 9660 image files on hard disks (see "Disk images on hard disk" on page 209). Using NT file compression, multiple images can be copied to a single hard disk.

Table 1: Supported jukeboxes

Manufactor	Type	Discs	Drives	Details see page
ASM	ASM CDR????	100-1563	1-44	164
Cope	Tower	6-7	6-7	165
Cygnet	Infinidisc/Infiniwriter	250/500	2, 4, 6, 8	166
Cygnet	ID100	100	1-4	167
Denon	DRD-1408	200	2	168
DISC	D???	238-1478	18-48	169
DISC	DA***.*			170
DISC	CD-CHG DJ-200/600	200/600	2-6	171
DSM	Terastore	28-1645	1-x	172
ELMS	DVL	100	4	173
Grundig	GMS 1035	35	2	174
Grundig	GMS 3200	200	1-x	175
Grundig	GMS 3280	280	6	176
HP	WORM/MO	any	any	177
Hyundai	HAS-550			178
JVC	MC-* CDROM Library	200/600	2-6	179
Kodak	ADL 100	100	1	180
Kodak	ADL 150	150	4	180
Kodak	CDL 144	-162	-4	181
Kubik	CDR240M	240	2-4	182
MDI	CD150	150	4	183
Nakamichi	MCD-1020	7	1	184
Nakamichi	MJ-4.8s	4	1	185
Nakamichi	MJ-5.16si	5	1	186
NSM	CDR100XA	100	1	187
NSM	CDR100Rec	100	1	187
NSM	Mercury 20	150	2	189
NSM	Mercury 31	150	4 (3/1)	189
NSM	Mercury 40	150	4	189
NSM	Mercury 20s/31s/40s	150	4	191
NSM	Satellite	-130	1-	192
Panasonic	LF-J50/100/200	50/100/200	2-4	193
Pioneer	DRM-6??x	6	1	194
Pioneer	DRM-1804x	18	1	195
Pioneer	DRM-1004x	100	2-4	196
Pioneer	DRM-5004x	500	2-4	197

Manufactor	Type	Discs	Drives	Details see page
Plasmon	CD150J	150	4	200
Plasmon	D-Series	120/240	2/4/6	199
Plextor	MegaPlex	200	2	201
Plextor	PX-J2200	200	2	201
Smart and Friendly	CDJ 7004	7	1	202
Smart and Friendly	CDJ 4008	4	1	203
Sony	CDZ-R360	360	2	204
Sony	CDL-2?00-??	125 (225)	2-4	205
any	single drive	1	1	208
-	image on hard disk	-	-	209

Table 2: Supported CD recorders

Manufactor	Type	IFS*
HP	SureStore 4020i	yes
JVC	XR-W1001	
JVC	XR-W2001	
Kodak	PCD 225	
Matsushita	CW-7501	
Matsushita	CW-7502	
Philips	CDD-522	yes
Philips	CDD-2000	yes
Philips	CDD-2600	yes
Philips	(others)	yes
Pinnacle	RCD-1000	
Pioneer	DR-R504X	yes
Plasmon	CDR-480	
Plasmon	CDR-4240	
Plasmon	CDR-4400	yes
Plasmon	RF4100	
Plextor	CD-R PX-R24CS	yes
Plextor	CD-R PX-R412C	yes
Ricoh	RO1060C	
Ricoh	RO1080G	

Manufactor	Type	IFS*
Ricoh	RS-9200CD	
Ricoh	RS-9200GD	
Sony	CDU920S	
Sony	CDU924S	
Teac	CD-R50S-000	
Teac	CD-R55S	
Yamaha	CDE/CDR100	yes
Yamaha	CDR400(c,t)	yes
*IFS = Incremental file system, allows track-by-track writing of disks		

Due to upward compatability of the SCSI command set other recorders of the listed manufactors should also work with `cdglow`.

2 Installation

2.1 Introduction

This chapter covers the installation of iXOS-JUKEMAN on Windows NT and UNIX hosts. A full installation is performed on the jukebox server. In addition the administration client can be installed on other hosts to allow remote administration of the iXOS-JUKEMAN server.

Note: To set up and use the software right after installation, the devices to be controlled by iXOS-JUKEMAN should be connected and operational.

To exploit all the features of the software, license keys have to be entered *after* the installation. Otherwise the software will run in demo mode. After the installation, read the section “Set up license keys” on page 31.

2.2 Requirements

2.2.1 Supported operating systems

iXOS-JUKEMAN runs with the following operating systems:

- AIX 4.*
- DEC UNIX 4.0 (Alpha processor)
- IRIX 6.2, 6.4
- HP-UX 10.*
- Windows NT 3.51, 4.0 (with Alpha or Intel processor)
- Solaris 2.4, 2.5, 2.51, 2.6

An updated list can be ordered from **support@europe.jukeman.com**.

2.2.2 System requirements

- At least 32 MB memory, 64 MB recommended for burning disks.
- At least one SCSI controller. It is recommended to have a recorder connected to a separate SCSI controller if you wish to use a recorder with `cdglow`. For best results we recommend using Adaptec controllers (2940, 3940). **There may be SCSI problems with IBM and Mylex RAID controllers on Windows NT.**
- 20 MB hard disk space for the software. The directory cache size depends on the number of disks and files on each disk. To evaluate disk space requirements see “Set up caches and buffers” on page 36.
- A hard disk buffer must be configured before using the incremental file system. The buffer size required depends on the amount of data to be held in the buffer.
- Sufficient free space for the database “`volumes`” in which iXOS-JUKEMAN stores information about the disks. About 2 kB per licensed disk will be needed.
- TCP/IP must be configured on the iXOS-JUKEMAN server (IP address etc.). The protocol used for the network is irrelevant.

2.3 Windows NT installation

The installation program for Windows NT copies files to a directory on your hard disk, installs iXOS-JUKEMAN services and sets up the registry entries.

Before you begin installation, log on as administrator or a super user.

Start the program **`jukeman.exe`** on the iXOS-JUKEMAN CD-ROM. The following dialog will open:

In this set-up dialog, specify where you want to install the iXOS-JUKEMAN user programs. (The default is `%SystemDrive%\jukeman`.)

Now select the portions of the software you want to install:

Install Jukebox File System

To install the administration service and the jukebox daemon, which controls jukeboxes and supports both file system and administration service. The writer and the `iso9660` formatter will be installed automatically.

Install Administration Client

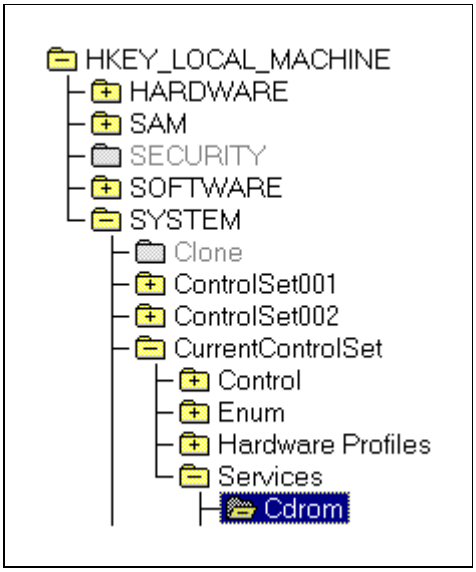
To install a graphical user interface with which you can locally or remotely administer the iXOS-JUKEMAN server.

Disable Autorun Feature

A feature of Windows NT is to run special files on CDs automatically (e.g., to start an installation). However, this autorun feature can lead to prob-

lems and undesired effects on the JUKEMAN server. A checkmark by **Disable Autorun Feature** will assure that no autorun files will be started on the server. This feature can also be configured in the registry (`regedt32.exe`):

autorun feature



You will probably want to install all these items. If you install the administration client on other hosts the iXOS-JUKEMAN server can be administered through the network. The administration client will also run on Windows 95.

The iXOS-JUKEMAN services (iXOS Jukebox Daemon and iXOS Admin Server) will be started automatically after reboot.

Note: The startup behavior of these services can be configured with the "Services" dialog from the NT "Control Panel".

Click **[OK]** to start the installation. The installation program creates an iXOS-JUKEMAN folder containing program icons.

When installation is complete, reboot your system.

To set up the software see "Setting up iXOS-JUKEMAN" on page 29.

2.4 UNIX installation

Before you begin installation log on as root.

All the files needed for installation can be found on the iXOS-JUKEMAN CD-ROM or downloaded from the world wide web (see page 9). To download, follow the instructions on the web pages. Important files to be downloaded are the operating system-specific file and the file with configuration examples ('`examples.tz`').

The following table lists the operating system-specific files. These files are compressed 'tar' files. To avoid problems with Windows programs the extension of the files is `.tz` rather than `.tar.Z`.

Operating system	**File name**
AIX 4.*	`aix.tz`
DEC UNIX 4.0	`dec.tz`
HP-UX 10.*	`hpux.tz`
IRIX 6.2, 6.4	`irix.tz`
Solaris 2.4, 2.5, 2.51, 2.6	`solaris.tz`
(all)	`examples.tz`

If all files needed are available, installation may begin:

You will need about 4 MB of disk space for installation.

Create an empty target directory (e.g, `jukeman`):

```
mkdir <path>/jukeman
```

If you are using an iXOS-JUKEMAN CD-ROM, you will find the software packages for certain UNIX platforms in a subdirectory called '`os`' or '`unix`'. For instance there is a file called `solaris.tz` for the Solaris platform. Change to the source directory where the software package for your UNIX platform is located.

Copy the file (e. g., `solaris.tz`) and the examples file (`examples.tz`) to the target directory, rename the file to `solaris.tar.Z` (and `examples.tar.Z`) and decompress the files. This can be done with the following commands:

```
cp solaris.tz <path>/jukeman/solaris.tar.Z
cp examples.tz <path>/jukeman/examples.tar.Z
cd <path>/jukeman
zcat solaris.tar.Z | tar xvf -
```

Now you have all the files except license files, device descriptions and configuration.

If you wish to **update an earlier version**, copy the old configuration files (`server.cfg`, `*.dev`, `*.sav`, `*.lic`) and the database `volumes` to the new target directory. For a first installation it is recommended to extract template configuration files by issuing the following command:

```
zcat examples.tar.Z | tar xvf -
```

The database `volumes` will be created by `cdnfsd` automatically. Now that you have all the relevant files, the tar files can be deleted:

```
rm *.tar.Z *.tz
```

Note that for further installation all jukeboxes connected to the host need to be operational in order to install the SCSI driver properly.

To install the SCSI driver:

```
jmsetup
```

This command installs both the SCSI driver if needed, and the iXOS devices paths. Some older HP machines (800 series) may tell you to install the SCSI pass through kernel driver (see `man scsi_pt(7)`).

Since the `cdnfsd` program runs only under root id and the performance of `cdglow` is much higher under root id, change the owner to root and the permissions to suid:

```
chown root cdnfsd cdglow
chmod u+s cdnfsd cdglow
```

For HP-UX 10.* change the permissions for both the `cdadm` and the `inquiry` command:

```
chown root cdadm inquiry
chmod u+s cdadm inquiry
```

To set up the software, see “Setting up iXOS-JUKEMAN” on page 29.

If you just want to use the writer software you do not have to set up anything else. The software is described in the section “Burning disks” on page 133. If you have already received a license key for the writer software from your distributor, enter it in the file `writer.lic` as described in “How to enter license keys” on page 34.

3 Setting up iXOS-JUKEMAN

3.1 Introduction

iXOS-JUKEMAN must be set up before you can start and use the software. The following sections will tell you how to:

- **Set up license keys.** This section may be skipped if you have no license keys and want to test the software in demo mode.
- **Set up caches and buffers.** Setting up a directory cache will improve the overall performance of the server. Setting up this cache pays off as soon as a new jukebox is attached to the server.
- **Set up devices.** This sections tells you how to configure device description files which allows you to access the disks in your jukeboxes, drives and so on.
- **Set up views.** Here you learn how to set up the presentation of the iXOS-JUKEMAN file system. By default iXOS-JUKEMAN sets up two views, which export all disks in both PC format and Rock Ridge format.
- **Intregrate iXOS-JUKEMAN into the network.** This section describes how to set up both the server side and the client side of iXOS-JUKEMAN, allowing all clients to access the disks they need.

There are two ways the set-up can be performed:

1. With the **GUI** on **Windows NT**. The administation client must be started. This can be done from the NT “Start” menu (**[START]-[PROGRAMS]-[IXOS-JUKEMAN]-JUKEMAN ADMINISTRATION** or by running “`jukeboy`” from the JUKEMAN directory). The corresponding sections are entitled **GUI**.
2. By editing configuration files on **Windows NT** and **UNIX**. Sections with this set-up method are entitled **CLI** (Command Line Interface). The configuration files can all be found in the JUKEMAN directory.

Both methods are equivalent. Any difference will be noted in the text.

3.2 Set up license keys

3.2.1 Why do I need license keys?

Valid license keys allow you to take advantage of all the features of iXOS-JUKEMAN.

There are separate license files for the server and the writer software, which specify a possible license time-out restriction and the number of disks the server will be allowed to handle. The license files are:

- `server.lic` contains the number of supported physical disks.
- `writer.lic` allows the writer software to burn CDs, PDs, WORMs and MOs on the JUKEMAN server.

demo mode

There are no special demo versions of the software. The server and the writer will run in a demo mode if the license key in the relevant license file is invalid. In demo mode the server will support five disks for two hours. After two hours it will stop, but can be started again.

The writer software `cdglow` and the incremental file system will burn up to 128 MBs in demo mode. However, creating a hard disk image from a disk is not limited.

Note: When the server starts up, iXOS-JUKEMAN will log information on the license files in the file "`logfile.txt`". This file is described in the section "Log file logfile.txt" on page 249.

3.2.2 How do I get license keys?

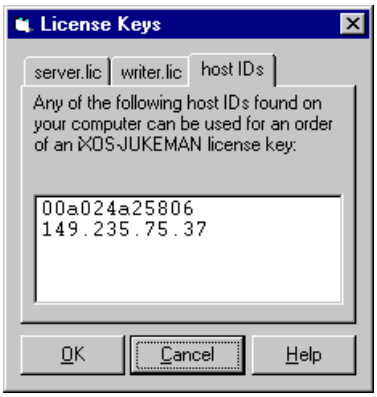
There are separate license keys available for server and writer software. The server license depends on the number of supported disks. The license key is bound to these capabilities and a host ID. The ID can also be the IP address. The host ID can determined with the following commands:

AIX:	`/bin/uname -m`
DEC UNIX:	`/sbin/ifconfig ln0` IP address
HP-UX:	`/bin/uname -i`

IRIX:	`/sbin/sysinfo -s`
NT:	either the IP address in the result of '`ipconfig`' or the network adapter address, which can be determined by the last entry of the "Workstation active on" line of the result of `net config workstation`.
Solaris:	`/usr/ucb/hostid`

The host ID also appears at the top of the file named `logfile.txt` in "Your key for a license order:".

For an NT host with more than one IP address or network adapter any of the addresses can be used to obtain a valid license key. The NT host IDs can also be determined with the GUI by selecting **[SERVICE]-LICENSE KEYS**:



Time-out licenses

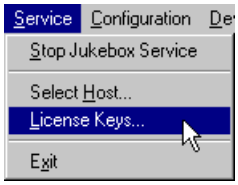
You can obtain a 30-day evaluation license for a limited number of disks from our web server **http://www.jukeman.com**. If you encounter any problems, send mail to **support@europe.jukeman.com**.

Unlimited licenses

You can obtain full licenses from your vendor. If you do not know where to buy iXOS-JUKEMAN, see **http://www.jukeman.com** or contact **pre-sales@jukeman.com**.

3.2.3 How to enter license keys

GUI **Windows NT**



Select **[SERVICE]-LICENCE KEYS**:

In this dialog the license keys for the server and the writer are entered independently.

Major Version Number: Should be '2' for all versions 2.xx of iXOS-JUKEMAN.

Server license for _ volumes: Enter the number of supported physical disks.

License expires...: For an evaluation license, deselect 'never' and enter the time-out date.

License key is: Enter your 8 character license key.

You can see '**invalid**' change to '**valid**' as soon as your license key is typed in correctly. Confirm the changes by clicking **[OK]**. If the service was running, it needs to be restarted to reflect the changes. In this case a dialog will open and tell you just that. Click **[YES]**, to restart the service.

CLI **UNIX, Windows NT**

Open one of the following files `server.lic` or `writer.lic` with a text editor. The default files look like this:

`server.lic:`

unlimited license

```
version=2
volumes=250
license=justdemo
```

time-out license

```
version=2
volumes=250
timeout=1997/06/13
license=justdemo
```

`writer.lic:`

unlimited license

```
version=2
writer
license=justdemo
```

time-out license

```
version=2
writer
timeout=1997/06/13
license=justdemo
```

The settings printed in **bold** must be changed to match your license.

Replace '**justdemo**' by the relevant 8 character license key. In `server.lic` replace **250** by the number of supported disks. The order of the parameters must not be changed.

Save the changed license file. The new settings will be active when the server is restarted.

3.3 Set up caches and buffers

Caches speed up access to frequently used data. With iXOS-JUKEMAN you can set up both a directory and a data cache. Caches on your hard disk are not required to run iXOS-JUKEMAN, since it always holds a directory and data cache in RAM. For a large number of disks, however, it is recommended to set up at least a directory cache.

In addition, iXOS-JUKEMAN supports incremental writing to disks with the incremental file system. To use this feature, iXOS-JUKEMAN needs a buffer on your hard disk to temporarily store the data to be written to the disks. Using the incremental file system, it is almost as easy writing to optical disks as writing to the hard disk.

Set-up of	**Recommendation**
directory cache	yes
data cache	no (see *"Note"* on page 40)
IFS buffer	required for incremental file system only

3.3.1 The directory cache

iXOS-JUKEMAN maintains a directory cache for all controlled disks. It stores the directory structure and file name information in the cache. This can improve performance, because a file search can be conducted without having to insert disks into drives of a jukebox. The cache is cyclic: old entries are dropped, when the cache is full and new entries arrive. The default size for the directory cache is 1 MB in RAM. Depending on the number of disks controlled by iXOS-JUKEMAN you may want to set up a larger hard disk-based directory cache.

Because of sophisticated hash techniques, the directory cache offers exceptionally high performance.

The cache is filled dynamically, when clients access the server and request non-cached directories and files. Internally the caching occurs under the following conditions:

1. If a directory is accessed and it is not found in the cache, then it is filled into the directory cache.

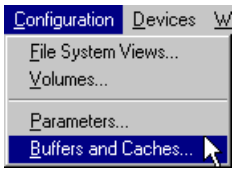
2. If a new disk is inserted into a drive, iXOS-JUKEMAN checks to see if the root directory is in the cache. If it is not, the parameter `autodc` (see section “Server parameters” on page 123) determines what happens next. If set to 0, no caching is performed. If set to 1, caching is performed only if a permanent hard disk directory cache is configured. When set to 2, JUKEMAN will cache the entire directory structure.

When setting the size of the directory cache, keep the following in mind:

1. Each file requires 25 bytes plus the length of the file name.
2. Each directory requires 100 bytes plus the length of the directory name.
3. Administration of the cache consumes approximately 15% of the space, so you have to add about one sixth of the totals of items 1 and 2 to the sum.
4. If a disk contains extensions in Rock Ridge format, you must allow for the additional space occupied by the rock ridge name.

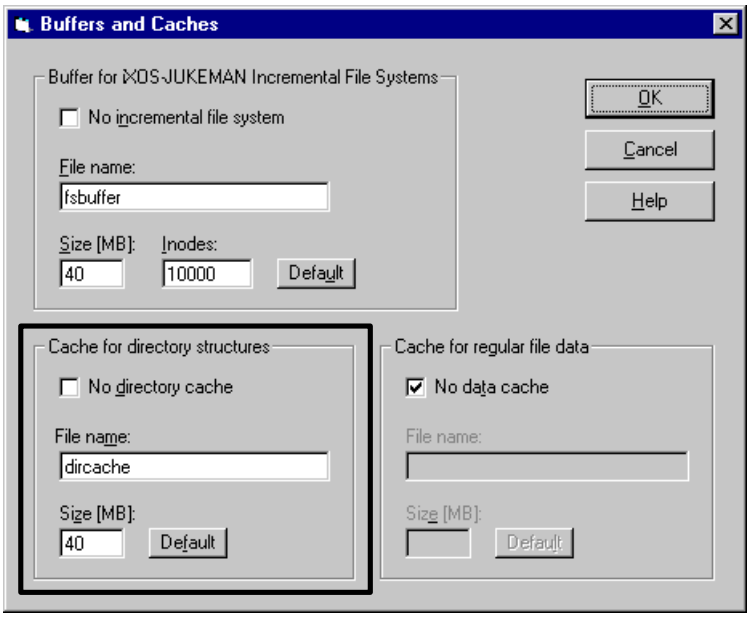
3.3.2 How to set up the directory cache

GUI **Windows NT**



Select **[CONFIGURATION]-BUFFERS AND CACHES**:

If the server is running, iXOS-JUKEMAN asks you to shut it down. Click **[YES]** to stop it.



1. Deselect **No directory cache**.
2. Click **[DEFAULT]** for the default settings.
3. Enter a file name for the cache in the **File name** field. Please note that for security reasons the cache file can be created in the JUKEMAN directory or its subdirectories only. If you do not specify a file name, the size of the RAM directory cache will be changed.
4. Enter the cache size in megabytes in the **Size** field.
5. Click **[OK]**.
6. If the server has been stopped, a dialog appears to restart it.

CLI **UNIX, Windows NT**

Note: The file `server.cfg` must be modified for this task. A detailed description of the file can be found in “Configuration file server.cfg” on page 237. It is recommended that you make a backup copy of the file before editing it.

1. Open the file “`server.cfg`” with a text editor.
2. Find the section “`dircache { ... }`”. If it does not exist, add the following lines to the end of the file:

```
dircache {
     file { <name> }
     size { <size> }
}
```

3. Enter the file name for the cache instead of *<name>* (e. g., `dircache`). If you do not specify a path, the cache will be created in the JUKEMAN directory. This is recommended for security reasons. If you want to change the size of the RAM directory cache remove the line “`file { <name> }`”.
4. Enter the cache size in megabytes instead of *<size>*.
5. Save the file.
6. The changes become active the next time the server is started.

3.3.3 The data cache

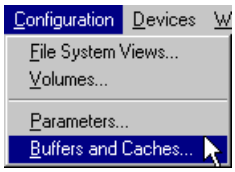
In addition to the directory cache, iXOS-JUKEMAN allows to set up a cyclic cache for regular data. All data accessed will also be stored in this cache. When the data is accessed again, it can be read directly from the data cache, so the disk does not have to be accessed. Being cyclic like the directory cache, when the data cache gets full old data will be replaced as soon as new data is accessed.

The default size of the data cache is 2 MB in RAM. A data cache on the hard disk can also be set up as a cache file. The name and maximum size of this file is entered in the configuration file `server.cfg`.

Note: Setting up a data cache on hard disk does not always guarantee a higher data rate, because quite frequently when data is accessed by the clients, it has to be written to the hard disk as well. Depending on the size of the data cache and the number of accesses the data rate may as well decrease.

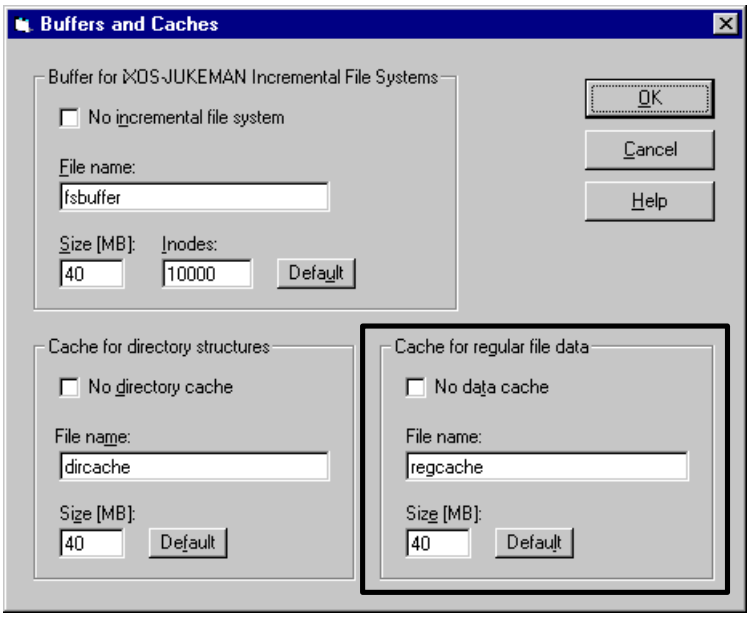
3.3.4 How to set up the data cache

GUI **Windows NT**



Select **[CONFIGURATION]-BUFFERS AND CACHES**:

If the server is running, iXOS-JUKEMAN asks you to shut it down. Click **[YES]** to stop it.



1. Deselect **No data cache**.
2. Click **[DEFAULT]** for the default settings.
3. Enter a file name for the cache in the **File name** field. Please note that for security reasons the cache file can be created in the JUKEMAN directory or its subdirectories only. If you do not specify a file name, the size of the RAM data cache will be changed.
4. Enter the cache size in megabytes in the **Size** field.
5. Click **[OK]**.
6. If the server has been stopped, a dialog appears to restart it.

CLI **UNIX, Windows NT**

Note: The file `server.cfg` must be modified for this task. A detailed description of the file can be found in “Configuration file server.cfg” on page 237. It is recommended that you make a backup copy of the file before editing it.

1. Open the file “`server.cfg`” with a text editor.
2. Find the section “`regcache { ... }`”. If it does not exist, add the following lines to the end of the file:

```
regcache {
     file { <name> }
     size { <size> }
}
```

3. Enter the file name for the cache instead of <*name*> (e. g., `reg-cache`). If you do not specify a path, the cache will be created in the JUKEMAN directory. This is recommended for security reasons. If you want to change the size of the RAM data cache, remove the line “`file { <name> }`”.
4. Enter the cache size in megabytes instead of <*size*>.
5. Save the file.
6. The changes become active the next time the server is started.

3.3.5 The IFS buffer (incremental file system)

With iXOS-JUKEMAN you can write a disk incrementally. After initialization you can copy or move files to a disk as easily as you would move them to your hard disk. The only difference is that you have to flush the IFS buffer to actually burn the buffered files to disk.

Note that you require a valid license key if the writer is to burn more than 128 MB to a disk.

To enable incremental writing, iXOS-JUKEMAN needs a global file system buffer on the hard disk. As soon as the buffer is configured, files can be copied to the disks. The buffer is configured by setting its size and the maximum number of files (“inodes”), that can be stored in the buffer.

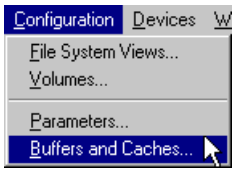
To determine the correct buffer size and number of inodes, keep in mind that each file or directory on an unfinished disk requires an inode and that the size should be sufficient to buffer all data you plan to transfer via the buffer.

Be sure to finalize all incrementally written CDs before you change the size of the incremental file system buffer (see “How to write disks incrementally” on page 156). Otherwise those CDs will be of no use.

As an alternative to a single IFS buffer iXOS-JUKEMAN 2.2 supports **several independent IFS buffers**. See “IFS with several independent buffers” on page 243.

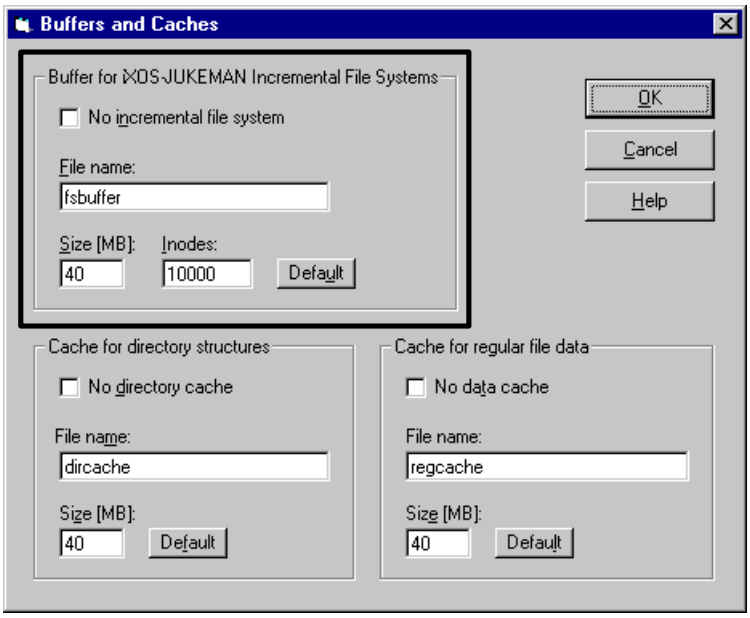
3.3.6 How to set up the IFS buffer

GUI **Windows NT**



Select **[CONFIGURATION]-BUFFERS AND CACHES**:

If the server is running, iXOS-JUKEMAN asks you to shut it down. Click **[YES]** to stop it.



1. Deselect **No incremental file system**.
2. Click **[DEFAULT]** for the default settings.
3. Enter a file name for the buffer in the **File name** field. Please note, that for security reasons the cache file can be created in the JUKEMAN directory or its subdirectories only.
4. Enter the buffer size in megabytes in the **Size** field.
5. Enter the number of files to be held in the buffer in the **Inodes** field.
6. Click **[OK]**.
7. If the server has been stopped, a dialog appears to restart it.

CLI **UNIX, Windows NT**

Note: The file `server.cfg` must be modified for this task. A detailed description of the file can be found in “Configuration file server.cfg” on page 237. It is recommended that you make a backup copy of the file before editing it.

iXOS-JUKEMAN 2.2 supports more than one IFS buffer (see “IFS with several independent buffers” on page 243). To set up a single IFS buffer do the following:

1. Open the file “`server.cfg`” with a text editor.
2. Find the section “`fsbuffer { ... }`”. If it does not exist add the following lines to the end of the file:

```
fsbuffer {
        file { <name> }
        size { <size> }
        inodes { <inodes> }
}
```

3. Enter the file name for the cache instead of *<name>* (e. g., `fsbuffer`). If you do not specify a path, the cache will be created in the JUKEMAN directory. This is recommended for security reasons.
4. Enter the cache size in megabytes instead of *<size>*.
5. Enter the maximum number of files instead of *<inodes>*.
6. Save the file.
7. The changes will be reflected the next time the server is started.

3.4 Set up devices

Each device controlled by the server requires a description file that specifies the device type, the SCSI address and properties of the drive, and the SCSI address or RS232 address of the robot (changing mechanism). The file also specifies which storage slots the server should use (default is all slots) and an optional file that stores the contents of the slots. A device can also be an ISO 9660 image on your hard disk, which is controlled the same way as ordinary disks.

The following sections explains how to set up SCSI devices for iXOS-JUKEMAN.

3.4.1 SCSI devices

This section summarizes the representation of SCSI devices by our generic SCSI driver.

Controllers, buses, IDs, and LUNs

A computer can use several SCSI controllers for multiple SCSI buses. Each SCSI bus has 8 IDs, named 0-7. Normally, each device on a SCSI bus requires a SCSI ID. Jukeboxes often need one ID per drive and one for the robot.

Warning: When you connect devices to the bus, be sure the new devices use IDs that are not being used by existing devices on the controller. If you use an occupied ID, you can damage your hardware.

Usually, the SCSI controller occupies ID 7, 0-6 are free for devices. SGI computers occupy ID 0, which means you can connect devices using IDs 1-7.

Each SCSI ID can be split into 8 logical unit numbers (LUNs). The advantage of LUNs is that a jukebox only has to use one SCSI ID, so you can attach more devices to a single bus.

SCSI devices and device names

In UNIX and NT, each device is represented by a device name, which is a path in the root file system tree.

SCSI devices in NT

NT hides the devices, but iXOS-JUKEMAN makes the names visible. The SCSI ID x on bus y of controller z is represented by the path `\\.\pzbytx`. So if you have a PCI bus and put Adaptec twin adapters AHA 3940 into bus slots 0 to 2, you will have 6 SCSI buses, and ID 4 on bus 1 of adapter 2 is represented as `\\.\p2b1t4`. Each enumeration begins with 0. Within an ID, LUNs are represented by appending a comma and the LUN. So `\\.\p2b1t4,1` is LUN 1 of SCSI ID 4. `\\.\p2b10t4` and `\\.\p2b1t4,0` are the same.

SCSI devices in UNIX

In UNIX, all devices are located in a subdirectory of `/dev`. iXOS-JUKEMAN creates directories such as `/dev/iXOS_SCSI0`, `/dev/iXOS_SCSI1`, and so on. Each directory represents a SCSI bus. Devices are represented by numbers. So if your bus is represented by `/dev/iXOS_SCSI1`, the device using SCSI ID 4 is represented as `/dev/iXOS_SCSI1/4`. If you use LUNs, add a comma and the LUN number. `/dev/iXOS_SCSI1/4,1` is LUN 1 of the device using SCSI ID 4 on bus `/dev/iXOS_SCSI1`.

SCSI device representation for iXOS-JUKEMAN	
NT	**UNIX**
\\.\p<*P*>b<*B*>t<*T*>,<*L*>	/dev/iXOS_SCSI<*B*>/<*T*>,<*L*>
(<*P*>=adapter, <*B*>=bus, <*T*>=SCSI ID, <*L*>=LUN)	

Windows NT: What is the path of MY Jukebox?

If you know which paths represent your devices, you can use the `inquiry` command (in the JUKEMAN directory) to verify the paths. For example, if you have a HP Sure Store Recorder 4020i that is connected to the only SCSI bus on an NT system using SCSI ID 5, the command

```
inquiry \\.\p0b0t5
```

will return something like

```
0000002 \\.\p0b0t5 is HP's CD-drive "C4324/C4325"
0000004 ProRevL 1.25, Firmware 04/15/96
```

The first line gives the drive type, the second line gives the firmware version.

```
0000000 Can't open \\.\p0b0t5
```

occurs if the driver is not running, the path is wrong, the device does not work, the termination is wrong, or the cable is bad. The device may also be unknown if it was not running when you booted the system.

If you do not know the paths that represent your devices, call

```
scsidevs
```

It returns a complete list of all known and working SCSI devices. If you add a new controller, you can use these commands to check whether the path names have changed.

3.4.2 Notes about some operating systems

Windows NT

How do I get rid of all the drive letters after connecting a new jukebox to the server?

When you connect a new jukebox or changer to the server, Windows NT maps each drive (and even each LUN) to a new drive letter. This is usually not desired. iXOS-JUKEMAN can change this behavior:

The easy way is to configure the jukebox to be attached automatically as soon as the server starts up. This is desribed in “Attach devices automatically” on page 102. After the **second** reboot of the server or, alternatively after a restart of the server **after** the device is set up and a **single** reboot of the server this problem is solved. A more sophisticated solution to this problem and the background of this process is described in "Frequently asked questions (FAQ)" on page 253.

Solaris

If you connect new devices to your Solaris host, either the SCSI driver must be reinstalled with `jmsetup` or the operating system must be stopped with `/etc/halt` and rebooted with `boot -r`. to prevent it from controlling the drives in the jukeboxes. For more information see `man vold` and `man vold.conf` or our FAQ (Frequently Asked Questions).

IRIX

In IRIX, the available SCSI IDs are 1 to 7. ID 0 is occupied by the controller. The removable media manager `mediad` may cause problems; see `man mediad`.

AIX

AIX uses a multiplex driver: You can use the devices but not see them. If you issue the following command:

```
ls -l /dev/iXOS_SCSI?
```

the output should look like this:

```
lrwxrwxrwx   1 root     system        13 Oct 22 14:22
     /dev/iXOS_SCSI0-> genscsi/scsi0
lrwxrwxrwx   1 root     system        13 Oct 22 14:22
     /dev/iXOS_SCSI1 -> genscsi/scsi1
```

This shows that the files are in fact symbolic links to our generic SCSI driver. You can use the `csh` expansion mechanism '{...}' to list the device paths created by the driver. For example the following command:

```
ls /dev/iXOS_SCSI{0,1}/{0,1},{0,1}
```

gives the following output:

```
/dev/iXOS_SCSI0/0,0 not found  (or does not exist)
/dev/iXOS_SCSI0/1,0 not found
/dev/iXOS_SCSI0/0,1
/dev/iXOS_SCSI0/1,1
/dev/iXOS_SCSI1/0,0
/dev/iXOS_SCSI1/0,1
/dmv/iXOS_SCSI1/1,0
/dev/iXOS_SCSI1/1,1
```

This shows that on the first SCSI bus (`/dev/iXOS_SCSI0`) on IDs 0 and 1 there are two hard disks or other SCSI devices which cannot be accessed by the operating system. The access restriction is on LUN 0 only, whereas LUN 1 normally replies to the command. Our `inquiry` command prints the following:

```
inquiry /dev/iXOS_SCSI0/0,{0,1}
/dev/iXOS_SCSI0/0,0: Bad file number
0000000 /dev/iXOS_SCSI0/0,1 is IBM's unknown "DORS-3"
```

The operating system is not as restrictive for CD drives and jukeboxes:

```
inquiry /dev/iXOS_SCSI0/6,0
0000000 /dev/iXOS_SCSI0/6,0 is IBM's CD-drive
"CDRM00203"
```

With the following you can find out which IDs and LUNs are occupied by which devices:

```
inquiry /dev/iXOS_SCSI{0,1,2,3,4}/{0,1,2,3,4,5,6},0
```

IDs and LUNs that are not used return “`SCSI-Error in 00 - TEST UNIT READY`”)

3.4.3 Serial lines

For several jukebox types, the robot is controlled through a serial line, which allows you to save a SCSI ID. For the NSM Mercury and Satellite jukebox, the serial interface allows the software to fully exploit the features of the jukebox and its ability to execute several movements in different states simultaneously. iXOS-JUKEMAN uses this parallel capability; even under high load the server can satisfy 14 client requests per minute for different CDs in a single Mercury or Satellite. Moreover, you can connect 16 NSM jukeboxes to a single serial line and all will be able to move simultaneously.

Each serial line is represented in the file system by a name specific to the operating system. The first two serial lines are:

AIX:	`/dev/tty0`	`/dev/tty1`
DEC UNIX:	`/dev/tty00`	`/dev/tty01`
HP-UX 10.*:	`/dev/tty0p0`	`/dev/tty1p0` oder `/dev/tty0p1` (je nach Rechnermodell)
IRIX:	`/dev/ttyd1`	`/dev/ttyd2`
NT	`com1:`	`com2:`
Solaris:	`/dev/ttya`	`/dev/ttyb`

IRIX For IRIX, also see the output of `man serial`.

AIX For AIX, special serial cables that are required are available from IBM. The messages “`alarm clock during tty_open`” and “`open(/dev/tty1) timed out`” in the log file indicate an incorrect serial cable.

The following section provides you with details of device description files for server set-up. These files are needed to access the connected devices

through the server. For Windows NT, this section may be skipped, as you can use the GUI to generate the device description files.

3.4.4 Device description files

A device description file contains lines in the format <*key*>=<*value*>. Some keys must be specified for all devices, others must be specified only for certain types.

Device type set-up

The most important parameter is the device type: `device=`<*type*>.

The supported Jukeboxes and their type are listed in the following table.

Table 3 - Device types for all supported jukeboxes

Type	Jukeboxes
cdr100	Kodak ADL 100, NSM CDR100XA and CDR100Rec
cygnet	Cygnet Infinidisc
cygnet_id100	Cygnet ID100
denon200	Denon DRD-1408
disc	DISC D???
disc_da	DISC DA***.*
disc_dj	DISC CD-CHG DJ-200/600
dsm	DSM Terastore
elms	ELMS DVL
grundig35	Grundig GMS 1035
grundig200	Grundig GMS 3200
grundig280	Grundig GMS 3280
hyundai	Hyundai HAS-550
image	Hard disk image of a disk or of an ISO 9660 file
jvc	JVC MC-* CDROM Library
kodak_cdl	Kodak CDL 144
kubik	Kubik CDR240M
mercury	Kodak ADL 150, NSM Mercury 20, 31 and 40
nakamichi	Nakamichi MCD-1020, MJ-4.8s, MJ-5.16si Smart and Friendly CDJ 7004 and CDJ 4008
pioneer6	Pioneer DRM-6??x
pioneer18	Pioneer DRM-1804x
pioneer100	Pioneer DRM-1004x

Type	Jukeboxes
pioneer500	Pioneer DRM-5004x
plasmond	Plasmon D-Series
plextor200	Plextor MegaPlex (also known as PX-J2200)
ps_lf_j	Panasonic LF-J-50/100/200
satellite	NSM Satellite
scsi2	Standard SCSI-2 jukeboxes
single	SCSI single drives
sony_cdl	Sony CDL-2?00-??
sony_cdz	Sony CDZ-R360
standard	NSM Mercury 20s, 31s and 40s, Plasmon CD150J, ASM Jukeboxen, Grundig M35
tower	Cope Tower
worm	WORM and MO jukeboxes

Drive set-up

Each device description file must also contain a line such as `drive=`*<path>* for each existing drive, where the specified *<path>* is the name of the drive. The way the drive names are represented is described in “SCSI devices and device names” on page 49.

Please note:

The order in which the drives are listed is important and should resemble the view of the jukebox. Check your jukebox manual to determine which SCSI ID belongs to which drive.

You have four choices for the `drive=` lines. They differ in what character, if any, is to be added after the ‘=’:

1. If the drive is either not available or defective, declare this using the syntax **`drive=!`**. This inhibits all interaction of iXOS-JUKEMAN with the respective drive.
2. If you want to dedicate a recorder drive for writing disks, use the syntax **`drive=-`**. This specification allows the administrator to move disks into or from the drive, but file system requests will not use this drive. When the server is running, a drive can be locked and unlocked dynamically with `cdadm detach` *<device>* `[-d` *<drive>* `]` and `cdadm attach` *<device>* `[-d` *<drive>* `]`, respectively.

3. For a recorder drive you want to use for incremental writing from the file system, choose the syntax **`drive=*`** (see also “Burning disks incrementally” on page 151).
4. In all other cases, use the syntax **`drive=`**… .

For example, if you want the server to control a single drive that the generic SCSI driver presents as

`\\.\p0b0t4` or `/dev/iXOS_SCSI0/4`

the device description file should look like this:

Windows NT	**UNIX**
`device=single`	`device=single`
`drive=\\.\p0b0t4`	`drive=/dev/iXOS_SCSI0/4`

Similarly, a device description file for a pioneer6 changer would look like this:

`device=pioneer6`	`device=pioneer6`
`drive=\\.\p0b0t4`	`drive=/dev/iXOS_SCSI0/4`

Note: The drives in the device description file must be in the same order as the drive numbers in the jukebox and not in the order of the SCSI IDs. This applies to all jukeboxes with several SCSI IDs for the drives. See the documentation that came with the jukebox for more information on the drive order. The first physical drive of the jukebox must be the first drive in the device description file.

If the device type is a hard disk image, the drive parameter must be specified as one or more file names with an ISO image. (see “Disk images on hard disk” on page 209).

Slot set-up

Using the parameter `disks=<`*slots*`>` you can specify the slots to be used by the server. Sometimes it can be useful to use only some of the slots, for instance during set-up or when writable disks are stored in unused slots (reading empty disks can take some time depending on the drive). If the server is to use no slot at all, you can specify “`disks=-`”. In this case each disk has to be tested individually to be made known to the server, and the server forgets about the disks when it is shut down.

If not otherwise specified in the device description file, all slots will be used. You can specify a few disks by adding a line such as `disks=1-3` or `disks=1,2,4-6`.

The slots can be specified using the following syntax:

`-`	no slot
`7`	slot 7
`3,6,40`	slots 3, 6 and 40
`3-7`	slots 3 through 7
`2,20-45`	slot 2, and slots 20 through 45

Save file set-up

When the server attaches a device, it inspects only the specified disks. This is useful during installation when you start and stop the server often. If you do not want the server to inspect the disks upon each start-up, use the parameter `save=`<*savefile*> to specify a save file in which the server stores information about which disk resides in which slot of the jukebox. The server uses the file if present or creates it if it is not present.

```
device=pioneer6      device=pioneer6
drive=\\.\p0b0t4     drive=/dev/iXOS_SCSI0/4
save=p6.sav          save=p6.sav
```

We recommend to follow this naming scheme: Use the name of the device description file with the suffix `.sav` instead of the suffix `.dev`. This is also the default setting if the save file is specified as `save=*.sav`. If the name of your device description file is `p6.dev`, the corresponding save file should be `p6.sav`.

Please note, that if you change disks manually the state of the save file will be corrupted. Therefore, manual disk changes should only be done, if you are aware of how to regain a consistent state (see “`cdadm testcd <device> <list>`” on page 233).

Robot set-up

For the device types “nakamichi”, “tower”, “image” and “pioneer6” as well as “single” a robot need not be specified.

For all other device types, you must specify a robot: `robot=`<*rob*>. For most devices this is another SCSI ID. For “sony_cdz” and “pioneer18”, this is a LUN of the drive target. This is what a device description file for a “pioneer18” using slots 1-9 must look like:

```
device=pioneer18     device=pioneer18
drive=\\.\p0b0t3     drive=/dev/iXOS_SCSI0/3
robot=\\.\p0b0t3,1   robot=/dev/iXOS_SCSI0/3,1
disks=1-9            disks=1-9
```

For the “kubik”, the robot uses a serial interface. The device description file looks like this:

```
device=kubik         device=kubik
drive=\\.\p0b0t1     drive=/dev/iXOS_SCSI0/1
drive=\\.\p0b0t2     drive=/dev/iXOS_SCSI0/2
drive=\\.\p0b0t3     drive=/dev/iXOS_SCSI0/3
drive=\\.\p0b0t4     drive=/dev/iXOS_SCSI0/4
robot=com2:          robot=/dev/ttya
```

For NSM jukeboxes “cdr100”, “mercury”, and “satellite”, a single serial line controls up to 16 jukeboxes. An additional line `robid=<`*id*`>`, specifies the ID of the robot on the serial line. Each NSM jukebox has an ID from 0 to 15, which can be checked and set by the Mercury’s or Satellite’s menu, or directly in the CDR 100 (see NSM manual). A device description file for a Mercury 20 with Robot ID 7 should look something like:

```
device=mercury       device=mercury
drive=\\.\p1b0t0     drive=/dev/iXOS_SCSI1/0
drive=\\.\p1b0t1     drive=/dev/iXOS_SCSI1/1
robot=com2:          robot=/dev/ttya
robid=7              robid=7
```

Alternatively, the following syntax can be used to define the serial interface and robot ID in one line:

```
robot=com2:,7        robot=/dev/ttya,7
```

3.4.5 Further points to note

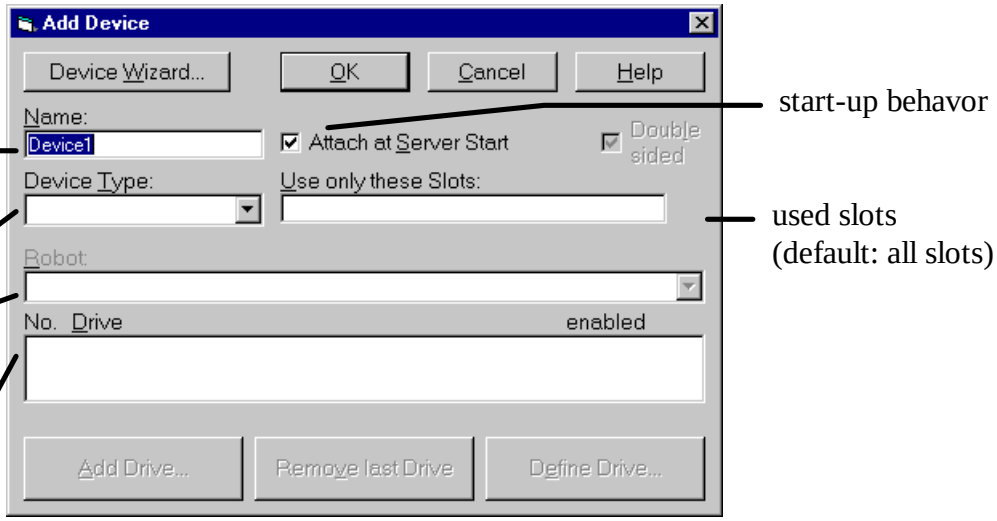
Make sure the device to be set up is connected properly and operational.

Note: The section “Supported jukeboxes” on page 163 provides you with more specific details about the supported jukeboxes with sample device description files, that can be used with minor modifications.

3.4.6 How to set up devices

GUI **Windows NT**

Select **[DEVICES]-NEW...** or click **[NEW...]**:



Entering the device name

Type a device name into the **Name** field. This name may be 8 characters long. This will also be the name of the device description file and the save file that will be created in the JUKEMAN directory (with the extension `.dev/.sav`).

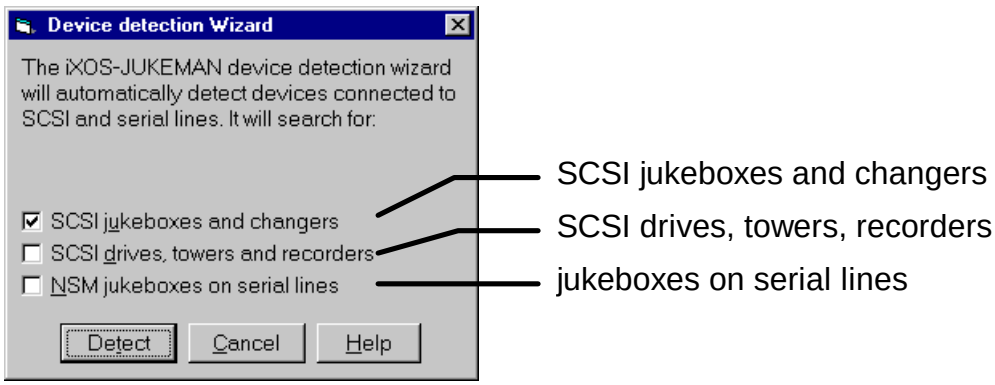
Setting up device type, drives and robot

For a quick set-up of the device description file, the 'Device Wizard' can be used. Many devices report their assigned SCSI IDs on request, allowing iXOS-JUKEMAN to add the drives to the device description file automatically.

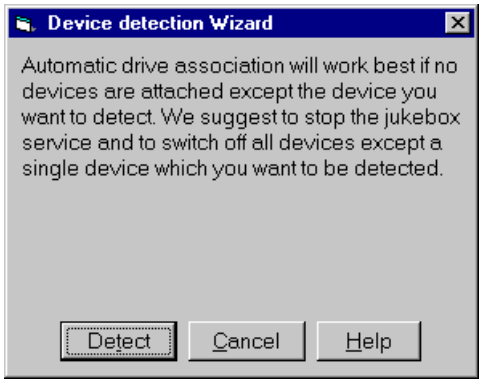
When running the automatic drive detection you should switch off all connected jukeboxes except the one to be configured.

Click **[DEVICE WIZARD]** to start the drive detection. The device wizard tries to find out as much as it can about the connected devices and to fill out the "**Device Type**", "**Drive**" and "**Robot**" fields. In case a device is not detected by the device wizard, these fields must be completed by hand.

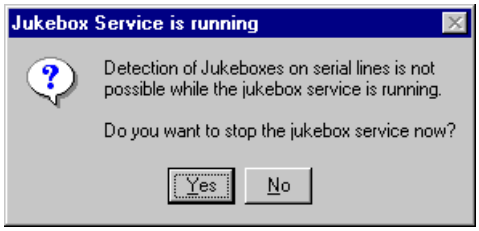
For example, all drives that could not be mapped properly appear as “undefined” in the drives list.



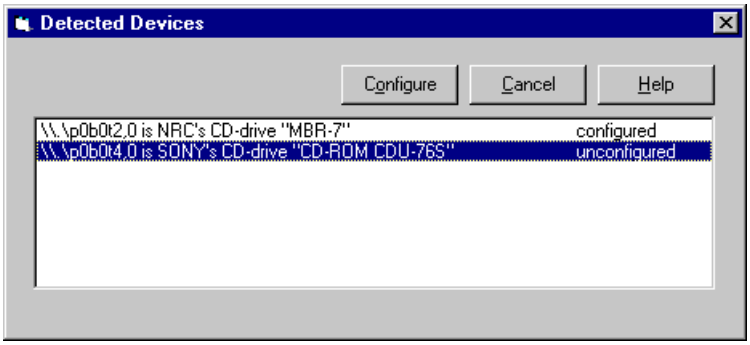
Select all device types you want the wizard to check for and click **[DETECT]**. Detection may take longer if the serial lines are searched for jukeboxes.



Click **[DETECT]** to go on.



If the serial lines will be searched for devices, the server needs to be stopped. Click **[YES]**.



From the list of detected devices, select the one you wish to configure and click **[CONFIGURE]**. The device wizard enters the values into the appropriate fields automatically.

Attach devices automatically

Select "**Attach at Server Start**" if the device should be attached automatically at server start-up.

Using only several slots

Enter the slots to be used in the "**Use only these Slots**" field. See page 56 for the syntax. If this field is left blank, all slots will be used.

Using double-sided WORMs/MOs

Select "**Double sided**" to configure a WORM or MO jukebox (otherwise this field is greyed out) with double-sided WORMs or MOs.

Add missing fields

If the device wizard failed to detect a device (e. g., if there is more than one SCSI device with the same SCSI ID attached to different buses), the following fields must be checked and completed:

Device Type

This is a list of all supported devices. Select your device. If your cannot find your jukebox in the list it may be a standard SCSI jukebox (see "Table 3 - Device types for all supported jukeboxes" on page 53).

Robot

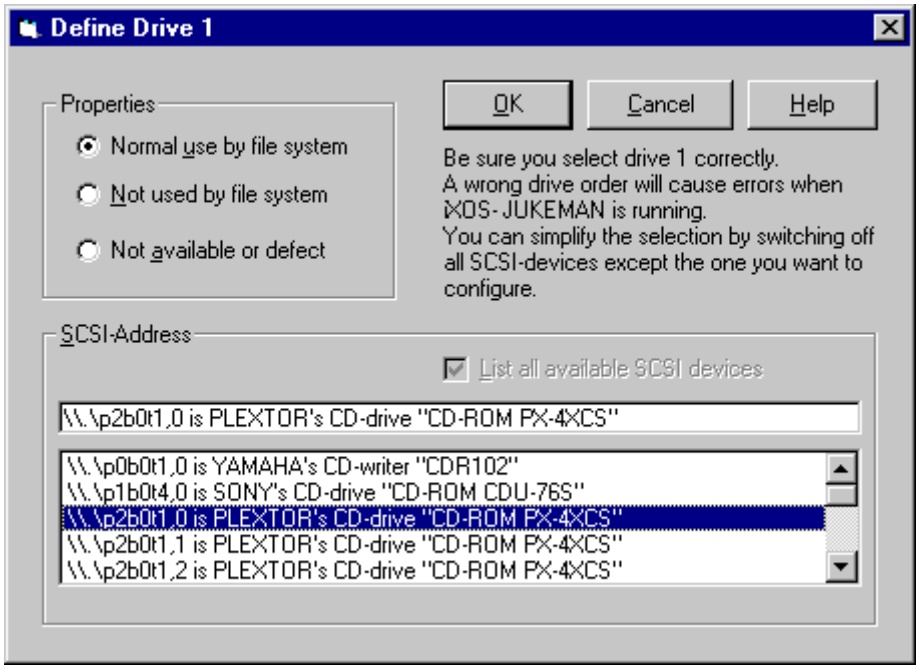
For most device types a robot must be specified. Whether it is a SCSI-ID (most jukeboxes) or a LUN of a SCSI-ID (like Pioneer 18

CD or Sony CDZ-R360), all possible selections will be listed in this field. If the robot is a serial line (as with Kubik- or NSM jukeboxes), the available serial interfaces and Robot IDs will be listed. See section “Robot set-up” on page 56. Select the appropriate robot from the list.

If no robot is to be specified (e. g., Pioneer 6), this field is greyed out.

Drives/Files

Each drive of a jukebox (or one or more files for a hard disk image) must be specified. Click **[ADD...]**, to select the drives or files. Click **[REMOVE LAST DRIVE]** to remove the last drive of the list. To change the properties of the defined drive, select the drive from the list and click **[DEFINE DRIVE]**.



The 'Define Drive' dialog lists all drives available, including a **defective/missing** entry for defect or missing devices. If for example the second drive of your jukebox is missing or not working, the second drive must be defined as **defective/missing** to assure the drives are accessed properly.

The drives must be added in the same order of the drives in the jukebox. In the “**Properties**” group in the upper left of the dialog, you can select the properties of the drive:

- **Not available or defect** is the same as **defective/missing**.

- **Not used by file system** can be selected for drives that will be dedicated to writing disks (see “Production of CDs in jukeboxes” on page 141).
- In all other cases select **Normal use by file system**.

These drive properties can be changed at a later time by double-clicking on a drive in the device definition dialog.

Click **[OK]** to define the drive. Click **[CANCEL]** to discard the changes.

Confirming or discarding the device set-up

Click **[OK]** to confirm the device set-up. If the server was stopped, it can be restarted with **[SERVICE]-START JUKEBOX SERVICE**. The devices marked as “**Attach at Server Start**” will be attached. All specified slots will be scanned for disks, and the directory structure will be stored in the directory cache. This may take a while, depending on the type of jukebox and the number of slots to be read.

Click **[CANCEL]** to discard the set-up.

CLI **UNIX, Windows NT**

Use a text editor to create a file "<*name*>.dev" in the JUKEMAN directory. Choose a <*name*> that gives a hint about the configured device (e. g., `mercury` or `tower`).

Enter the following lines in the file:

`device=`<*type*>

<*type*> is the device type of device to be configured. See "Table 3 - Device types for all supported jukeboxes" on page 53.

`drive=`<*path*>

Each drive must be specified with a line of this format. The order of the drive numbers is important, not the order of the SCSI IDs. The "=" may be followed by a "`!`", "`-`" or a "`*`". The meaning of these characters is described in "Drive set-up" on page 54.

<*path*> is the unique path of a SCSI ID for each drive. A detailed description of SCSI ID representation can be found in "SCSI devices and device names" on page 49.

`robot=`<*rob*>`[,`<*robid*>`]`

The parameter <*rob*> specifies the robot (SCSI-ID/LUN or serial interface) of the device. For NSM jukeboxes the parameter <*robid*> must be specified. See "Robot set-up" on page 56 for more information.

`[disks=`<*slots*>`]`

This line must only be specified, if you do not want to use all slots of a jukeboxe. The syntax for <*slots*> is described in "Slot set-up" on page 55.

`[save=`<*savefile*>`]`

If you enter this line, iXOS-JUKEMAN will store information on which disk is in which slot in the file <*savefile*> when the jukebox is attached for the first time. The name of the <*savefile*> should be the <*name*> of the device with the extension `.sav` (see also "Save file set-up" on page 56.

If all lines are entered, save the file "<*name*>.dev".

The section "Attach devices automatically" on page 102 tells you how to configure the server to attach one or more devices automatically at start-up.

If you have created a device description file named `device.dev`, the device can be attached with the command

```
cdadm attach device
```

The server will inspect the specified slots and present the disks to the views of the file system. iXOS-JUKEMAN will do this inspection if the devices are attached for the first time or if you attach a device that has no save file specified.

3.5 Set up views

The views concept allows you to set up the iXOS-JUKEMAN file system to your liking. This way a structure can be built to present all the disks clearly to network users.

Clarity

For instance, if you had hardware managing 700 disks and all these disks would be visible as subdirectories to a single root directory, the clients accessing the data might be presented with problems to find the desired disks. For this reason like in every file system a tree structure should be established, the disks being the leaves of this tree.

This allows not only the client access to be controlled but also the access to special disks will be made easier for the clients. A software developer may want to use only certain disks contraining PC products. It would speed up his or her work, if he or she could see just the disks he or she needs.

Name format

Moreover, different clients need different views to the file system. PCs usually need the file name format 8.3 with no consideration for upper or lower case, whereas some UNIX NFS clients prefer to use long file names embedded in the optional Rock Ridge extensions to the ISO 9660 file system standard. Or some clients may access all the disks, whereas other clients may only use a certain subset of the available disks.

iXOS-JUKEMAN supports a variety of views to the file system, which differ in the name format and the number of visible disks. “Figure 1” on page 70 presents some examples for different views in different name formats.

Name format	Meaning
`pc`	PC format (8.3)
`rr`	Rock Ridge extensions
`hs`	High Sierra format

The **PC format** is a modified High Sierra format, optimized for PC clients. The version number is suppressed and all file names will be converted to lower case. This is most important for PC clients that do the conversion to upper case characters themselves. If these clients received names in upper case the file names would be converted into strange, generic names. By default the file names will be in 8.3 format due to compatability resons for Windows for Workgroup clients. This behavior can be configured with the server parameter `fullvn` (see “Server parameters” on page 123). In

any case by explicitly renaming volume names any name up to 32 characters long can be chosen.

The **Rock Ridge** format displays file and directory names in ISO 9660 format, with regard to the Rock Ridge extensions (if available). These additions to ISO 9660 allow UNIX file names to be presented in ISO 9660 file systems. The Rock Ridge extensions may also contain UNIX file permissions.

In **High Sierra** format the file and directory names will be presented exactly as stored on disk. However, the version numbers that form an integral part of the ISO 9660 standard are useless for most clients. This format is only supported for completeness.

The views are defined in the file `server.cfg`.

Drive letter

On Windows NT, each view can be assigned a drive letter. This is optional, since views can also appear as subdirectories of other views, so the number of views is not limited by the number of available drive letters. To make a view available to the clients it is enough to share the appropriate directory or drive letter with the desired permissions.

A view is a path for UNIX clients, that can be mounted with NFS. There are three default views: There is the 'Root' view which can be mounted with '*<hostname>*`:/`' in the '`mount`' command. This 'Root' view contains two other views named '`views_pc`' and '`views_rr`'. '`views_pc`' is a view that contains all disks as subdirectories in PC format (8.3 format). '`views_rr`' also contains all disks, only the file names are in Rock Ridge format with long UNIX file names and permissions, if available. To mount this view on UNIX the '`mount`' command must include '`<hostname>:/views_rr`'.

These default views are defined in the file `server.cfg` and will be sufficient for most purposes. Feel free to add new views to this set-up, if needed.

Disk sets

Each view defined in `server.cfg` must contain a **disk set** specification. The disk set is a list of all the disk names that will be visible to the clients. For most views the disk set specification is `discs { * }`, in which case all disks are visible, but you can restrict the disk set by explicitly listing only those names that should be visible. Furthermore, invisible disks may be explicitly specified using the syntax `deny { }`. The curly brackets of both the `discs` and the `deny` section contain a list of blank separated disk names or `csh`-like meta characters as listed in the following table.

Table 4 - Disk set syntax

Char.	Meaning	Example	possible disks (e. g.)
?	any single character	ix?s	ixas, ixos, ixks..
*	any string	ix*s	ixs, ixmas, ix-otic...
[...]	list of alternatives	[abc]horn	ahorn, bhorn, chorn
[-]	range of characters	[a-m]horn	ahorn, ..., mhorn
[^]	restricted characters	[^d]*	all starting without “d”
{...,...,...}	alternative strings	{cow, mick}*	all starting with cow or mick

Summary:

- iXOS-JUKEMAN allows to set up a tree of views.
- Each view is represented as a directory.
- A view may contain subviews or any disk set of all the disks controlled by iXOS-JUKEMAN.
- Each view has a name format specifying how the contents of the view are presented to the clients.
- Each view can be shared to the clients with the operating system functions.
- Each view can be assigned an optional drive letter on Windows NT.

The following figure illistrates a view set-up, where view_1 and view_2 contain a selection of disks in pc format and rr format respectively. In contrast, view_3 contains two subviews (view_3a and view_3b). Subviews themselves may also contain views. In this case, however the subviews contain a selection of disks.

Please note, that view_3 is assigned a name format (rr). This name format will be inherited by the subviews, if nothing else is specified.

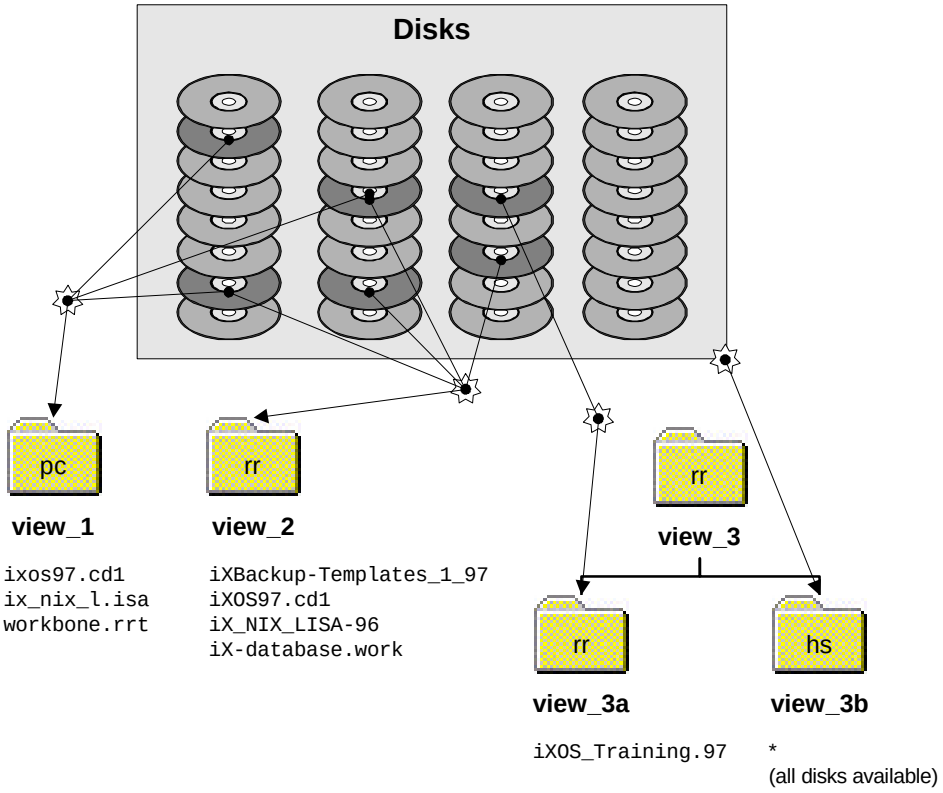


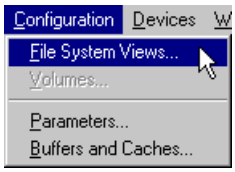
Figure 1 - Example for different views

In `server.cfg` the description for the above views set-up would look like this:

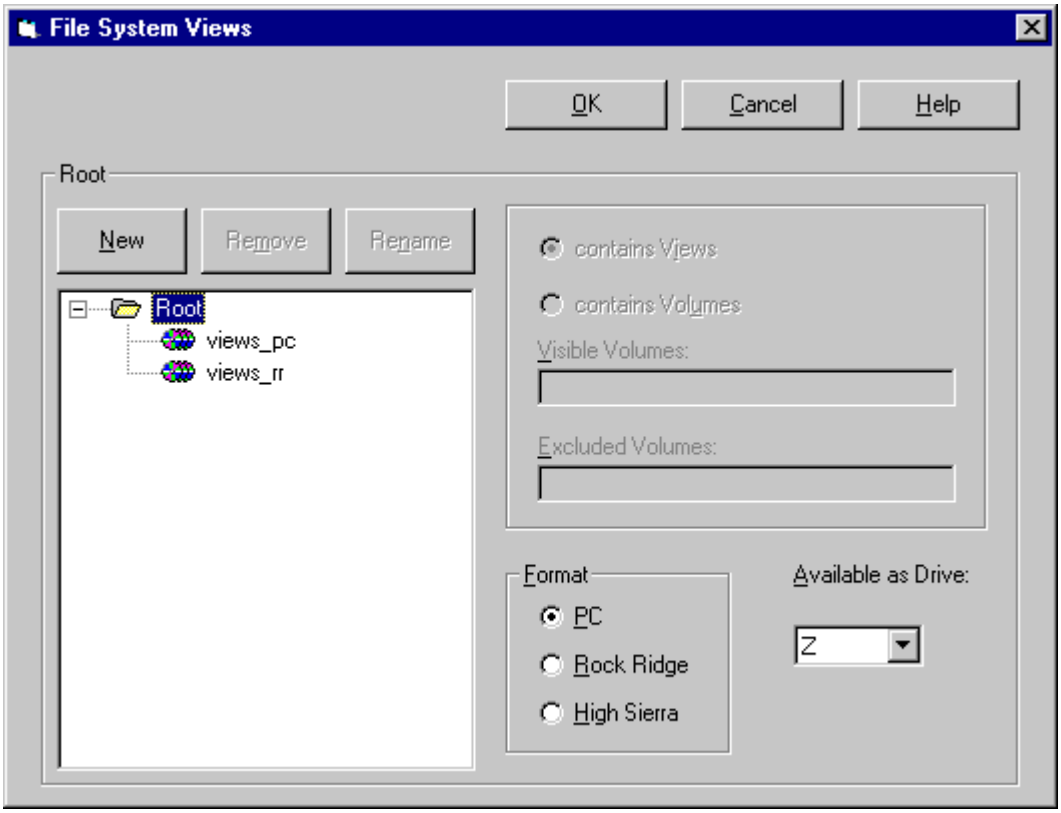
```
views {
  list { view_1 view_2 view_3 }
  roots {
    view_1 {
      discs { ixos97.cd1 usenix_l.isa workbone.rrt }
    }
    view_2 {
      discs { iX*}
      format { rr }
    }
    view_3 {
      format { rr }
      views {
        list {  view_3a view_3b }
        roots {
          view_3b {
            discs { * }
            format { hs }
          }
          view_3a {
            discs { iXOS_Train*  }
          }
        }
      }
    }
  }
}
```

3.5.1 How to set up views

GUI **Windows NT**



Select **[CONFIGURATION]-FILE SYSTEM VIEWS**:



The dialog shows the default views configuration. Using this dialog, new views may be added and configured, and existing views may be changed, renamed, or removed.

By default, the 'Root' view is mapped to the drive letter `Z:` and contains two subviews '`views_pc`' and '`views_rr`' in the PC name format and the Rock Ridge name format, respectively. '`views_pc`' is mapped to drive letter `X:` and '`views_rr`' is mapped to drive letter `Y:`.

Adding a new view:

Click on a view that has "**contains views**" selected. Click **[NEW]**. A new subview will be appended to the selected view. Now enter a name for the

new view and confirm with 'RETURN'. If a view is to contains subviews **contains views** must be selected for this view. Otherwise click **contains volumes**.

Renaming a view:

Select the corresponding view and click **[RENAME]**, or double-click the view. Enter a new name and press 'RETURN'.

Removing a view:

Please note that all subviews will be lost, if a view is deleted. Select the appropriate view and click **[REMOVE]**.

Assign a view a drive letter:

Select a view and a drive letter from the list **Available as Drive**. The drive letter can then be shared to the clients. If you select **[none]** wählen, the view will not be assigned a drive letter. **Please note, that executables cannot be started from these drive letters on the server.** To do this, the corresponding drive must additionally be mapped as a network drive on the server. This can be easily done with the Windows NT Explorer.

Setting up disk sets for a view:

The field **visible volumes** lists all disk names visible to the clients (all by default). The field **excluded volumes** lists all disk names, that will not be visible to the clients explicitly (none by default).

The specification can either be as meta characters in csh syntax (see “Table 4” on page 69) or explicitly by specifying a blank separated list of disk names. The **[CONTENTS]** dialog lists the disk names. Please note that the name must be entered in the correct name format. As opposed to the original disk name, the disk name in one of the three name formats is always unique.

Confirming or discarding the views set-up:

Click **[OK]** to confirm the views set-up.

Click **[CANCEL]** to discard the changes.

Please note:

If you change a view name the changes will not be active unless 'RETURN' is pressed. In some cases the **[OK]** button can only be clicked if you click somewhere else in the view tree (e. g. Root).

The administration client will save the views configuration in the file `server.cfg`.

CLI **UNIX, Windows NT**

Note: The file `server.cfg` must be modified for this task. A detailed description of the file can be found in “Configuration file server.cfg” on page 237. It is recommended that you make a backup copy of the file before editing it.

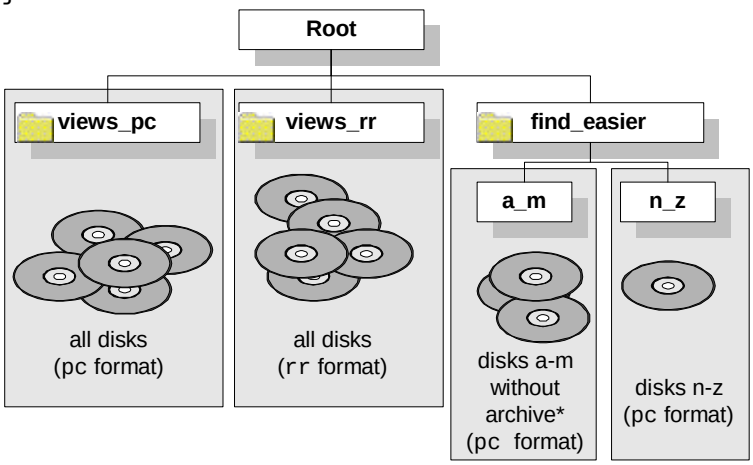
Open the file `server.cfg` with a text editor. The views are entered in the `views { ... }` section. The first parameter that must be specified is `list { }`, containing the names of all views on top level. Following this section is the parameter `roots { }`, containing all views names and their definition. Each view that contains subviews has its own `views { }` section (with its own `list { }` and `roots { }` section).

Each view that is to contain disks can have the following parameters:

Parameter	Value
`format`	Name format (`pc`, `rr`, `hs`). If this parameter is not specified the name format will be taken from the superordinate view. The default name format for views on top level is `pc`.
`discs`	The visible disks (`*` for all). See “Table 4 - Disk set syntax” on page 69.
`deny`	The excluded disks. See “Table 4 - Disk set syntax” on page 69.
`drive`	Drive letter under Windows NT Windows NT (will be ignored under UNIX). If not specified, the view will not be assigned a drive letter.
`label`	Label for the drive letter under Windows NT (Default: `JUKEMAN`, ignored under UNIX). May contain octal escape sequences like “\040” for a space.
`raw`	The `raw { 1 }` parameter selects a view format in which all disks are represented through the raw file system. You do not see the directories and files of the disks, but the full disk as a large file. The directory structure is explained in “Raw filesystem” on page 240.

Suppose you wanted to export the following disk set: A view `views_pc`, containing all disks for PC clients in the name format `pc`, a view `views_rr`, containing all disks for UNIX clients in the name format `rr` and a view `find_easier`, containing two subviews. One subview should contain all the disks starting with the letters `a`, `b`, `c`, ..., `m` in PC format and leave out all disks starting with `archive`. The other subview should contain all disks starting with the letters n-z:

```
drive { Z }
views {
  list { views_pc views_rr find_easier }
  roots {
    views_pc { format { pc } discs { * } drive { X } }
    views_rr { format { rr } discs { * } drive { Y } }
    find_easier {
      views {
        list { a_m n_z }
        roots {
          a_m { format { pc } discs { [a-m]* } deny { archive*} }
          n_z { format { pc } discs { [n_z]* } }
        }
      }
    }
  }
}
```



Please make sure that the number of opening brackets "`{`" must match the number of closing "`}`" brackets.

Save the file `server.cfg` when the set-up of the views is finished. On UNIX systems a directory must be created and shared for each view. How to do this is explained in the following section.

3.6 Intregrate iXOS-JUKEMAN into the network

Until now you have:

- installed iXOS-JUKEMAN
- entered license keys
- set up caches and buffers
- set up devices
- set up views.

This section teaches you the last few steps necessary to enable the clients to access the disks when the server is started.

3.6.1 The server side

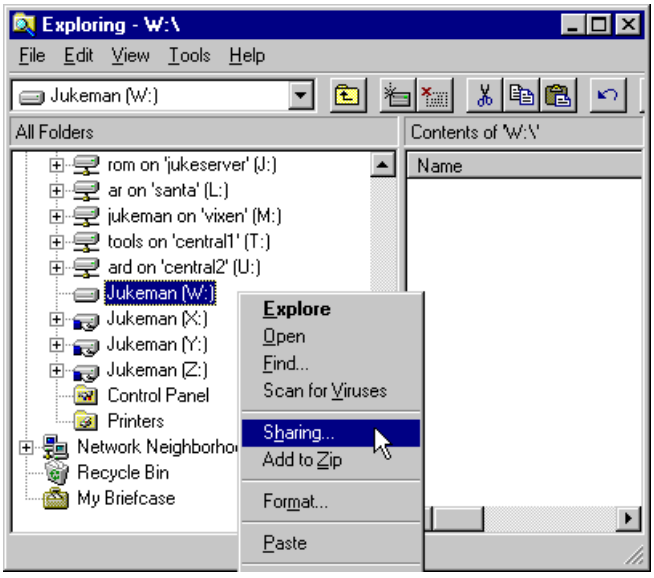
iXOS-JUKEMAN on Windows NT

The NT version of iXOS-JUKEMAN fits many different environments, including those that are mixed. It supports the NFS protocol, making it a true NFS server. UNIX clients simply mount it as they mount any network file system. Simultaneously, it presents all disks in a native file system for NT, which can be exported through all available protocols.

When the host boots up the iXOS-JUKEMAN server is started. Depending on the set-up of the views new drive letters appear for the views. If you did not modify the views configuration, the view `views_pc` will be mapped to drive letter `X:`. All disks of all connected devices will be displayed as subdirectories of `X:`. The drive letter `Y:` is the view `views_rr`. These two views will be shared to NFS clients automatically. PC clients can map `X:` and `Y:` as network drives provided these drive letters are shared. The Root view, containing both `views_pc` and `views_rr`, is mapped to drive letter `Z:`. It is also shared for NFS clients.

The disks can be accessed using the File Manager on the server. The drive letters or directories can be shared using the standard operating system functions (for administrators or super users).

All views you set up in addition must be explicitly shared. This can be easily done with the Windows NT Explorer. For example, if you configured a view which is mapped to drive letter `W:`:



In the Explorer, move the mouse pointer to the drive letter or directory you wish to share and click the right mouse button. Select **[SHARING]** from the pop-up menu. A dialog will pop up in which the drive letter or directory can be shared to the clients.

The following problems may occur in some cases:

- Sometimes a PC client has to deal with long file names (e. g., if a Rock Ridge view is mounted — the PC format view automatically converts the file names to 8.3) and cannot handle it. The solution to this problem is to rename the disks on the server. This can be done with the Windows NT Explorer, with a dialog of the administration client (see “Rename disks” on page 119) or with the command `cdadm rename`, described on page 225.
- Some CD software assume the root directory of a disk to be the same as the root directory of the drive, not taking into account the disk name as part of the path. As a result an installation may fail, because the appropriate files cannot be located. If this happens, share the corresponding subdirectory, map it as a network drive on the client PC and the installation will work.
- To execute programs on the server from the iXOS-JUKEMAN file system the disks have to be mapped as a network drive on the server. It is not possible to start these programs from the drive letters created with the views set-up.

iXOS-JUKEMAN on UNIX

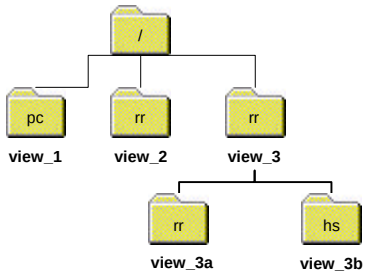
On UNIX iXOS-JUKEMAN cooperates with the standard NFS server. If no NFS is running, the set-up of the file system is finished. If NFS is running, you must set up the standard `rpc.mountd` daemon to cooperate with the server.

`rpc.mountd` monitors all file systems, including that of iXOS-JUKEMAN. If clients mount a network file system they ask `rpc.mountd` for a root file handle. This file handle is the key for any further requests to the standard `nfsd` or iXOS-JUKEMAN's NFS-Jukebox-Server. It is not possible to access the network file system without this key.

A root file handle can only be generated, is there is a root for the file system. Therefore, you must create an empty directory for each view. The hierarchy of the directories to be created must follow the hierarchy of the views. A subdirectory must be created for each subview. The views are normally `/views_rr` for the Rock Ridge format and `/views_pc` for the PC format. Any other file names are possible as well (although they do not comply with the standard `server.cfg` file and must be entered into this file accordingly).

If, for example, the views set-up corresponds to that of "Figure 1 - Example for different views" on page 70, the following directories must be created on the UNIX server:

```
mkdir /view_1
mkdir /view_2
mkdir /view_3
mkdir /view_3/view_3a
mkdir /view_3/view_3b
```



Next export a file system by telling `rpc.mountd` to give the file handles to clients. Most flavors of UNIX maintain a file `/etc/exports` (Solaris: see below) containing all exported file systems. You can simply add a line to this file containing just the character `/`. This will tell `rpc.mountd` to export all views of the server file system. Please note, that this will also cause all other directories of the server to be shared. It is recommended to export only the directories representing the views. To do this add the following lines to `/etc/exports`:

```
/views_pc
/views_rr
```

This will export the standard views to all hosts. You can also set the access permissions for certain hosts:

```
/views_pc -ro
/views_rr donald daisy garfield localhost
```

These lines will export `/views_pc` read-only to all hosts and `/views_rr` to the hosts `donald`, `daisy`, `garfield` and the server itself. Please keep in mind that the entry `localhost` is mandatory when specifying certain hosts.

On **Solaris** the exported file systems are listed in the file `/etc/dfs/dfstab`. To export the standard views, add the following lines to the file:

```
share -F nfs /views_pc
share -F nfs /views_rr
```

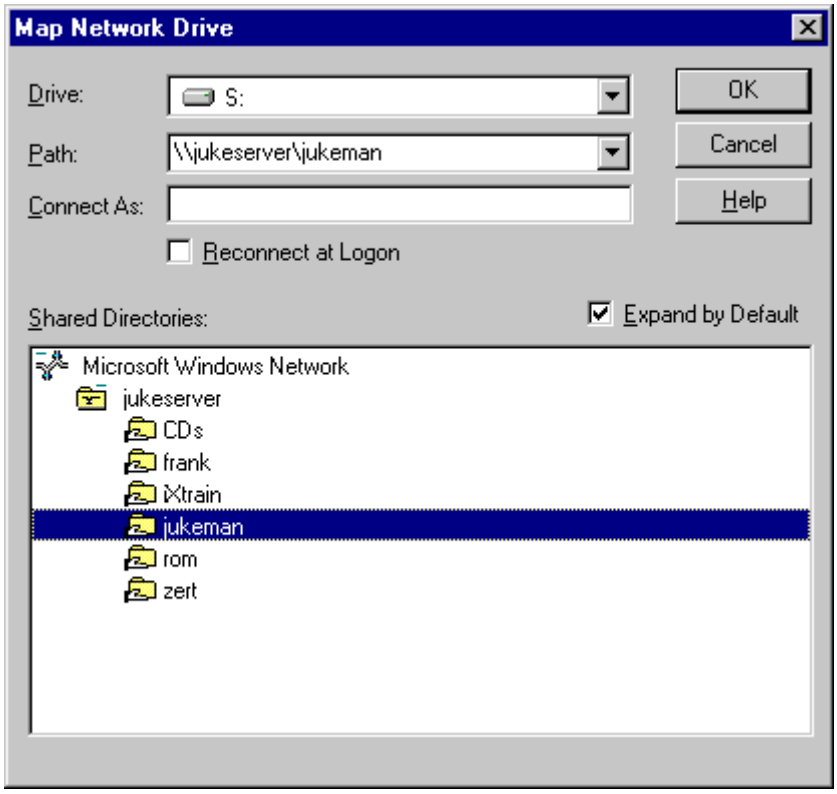
On some UNIX systems (Solaris and others), the change does not take effect immediately. On Solaris, call the commands `unshareall` and `shareall` to update the export list. Other UNIX systems offer the `exportfs` command. Use `exportfs -a` to export all file systems. Another method is to reboot the server. You can easily test if the directories are exported by using `showmount -e`. When you start iXOS-JUKEMAN, it tests whether the directories for all views are exported by requesting the file handle from `rpc.mountd`. It stops if it cannot get a file handle, and prints a message in the log file.

3.6.2 The client side

Windows clients

As soon as the directories representing the views are shared on the server it is possible to map the directories as network drives on the client PC.

For instance, the view `jukeman` on the server `jukeserver` is shared. To use it from a client, you can select **Map Network Drive** from the explorer:



NFS Clients (UNIX, NT)

NFS clients, e. g., UNIX clients, can mount the standard views

<*jukeman_hostname*>`:/views_pc`

or

<*jukeman_hostname*>`:/views_rr`

of a **Windows NT JUKEMAN server**.

For example:

```
mount -o timeo=99,retrans=14 jm_hostname:/views_pc /cds
```

The disks can then be accessed from the directory `/cds` on the client.

Both the UNIX and NT versions of iXOS-JUKEMAN support the NFS protocol. NFS clients can mount the file system of a **UNIX JUKEMAN server** as they mount any network file system, but they need to add some parameters for the `mount` command. For example,

```
mount -o port=4027,timeo=99,retrans=14,soft
<hostname>:/views_rr /cd
```

mounts the server's file system on the empty directory `/cd` of the client. Once this is done, all disks appear as subdirectories of `/cd`. The simple command `ls -l /cd` shows a list of all available disks.

Depending on the operating system, some versions of `mount` require additional parameters, e. g.,

`mount -F nfs -o port=..` or `mount -f NFS,port=....` See `man mount` for details.

The `port=4027` option tells `mount` that the NFS server uses port 4027 instead of 2049, which is used by the standard NFS daemon. This enables the server to coexist with the standard `nfsd` so clients can use both hard disks and jukeboxes on the server computer concurrently. NT does not include a standard `nfsd`; consequently, the NT version uses the standard port, and you do not need to specify the port number.

For some newer UNIX operating systems like Solaris 2.5, DEC UNIX 4.0 or IRIX 6.4 the `mount` command should include a further option '`vers=2`'. Without this option NFS version 3 would be used. iXOS-JUKEMAN supports NFS protocol version 2 only, so that the client would use version 2 anyway after negotiating with the server.

```
mount -o port=4027,timeo=99,retrans=14,soft,vers=2
host:/views_rr /cd
```

To understand the other options you need to be familiar with NFS clients: A user level application accesses a mounted network file system as if it were any local magnetic disk. The kernel of the client computer automatically generates NFS requests and waits for the answers, which in turn are used to satisfy the accesses requested by the application. But networks may drop a request or an answer. Therefore, the NFS client built into the client's kernel not only generates NFS requests, but also retransmits them if it does not receive a reply within a reasonable time.

The `timeo=99` option instructs the kernel's NFS client to retransmit a request if there is no reply after 99 tenths of a second (9,9 seconds). These retransmits are not visible to the users, except for messages such as "NFS `server not responding, still trying`". Short time-outs increase the network load because each disk move can cause several useless retransmits. Long time-outs are bad if a packet is dropped by an unreliable network and a user must wait until the kernel's NFS client retransmits the request. After each retransmit, the time-out value is doubled, up to a maximum of one minute.

The `retrans=14` option instructs the kernel's NFS client to automatically retransmit a request 14 times before it gives up and the file system access that caused the NFS request fails. It makes sense to specify a high value because if several clients access different disks located in the same jukebox, the server must move these disks, and the last client must wait a long time. You can avoid long wait times if you have enough jukeboxes and enable the server to distribute the load by duplicating the disks and spreading them over the jukeboxes. This enables you to build failure-tolerant archives with predictable short response times.

The `soft` option instructs the kernel's NFS client to give up after all retransmits. You can also specify `hard` angeben causing the client not to give up even after the last retransmit. If you specify an additional `intr` option the system call that caused the NFS request may be interrupted with a signal. If you specify `hard` without `intr` the only way to finish the system call is a server response.

If the `mount` command generates a "`no such file or directory`" message, make sure the `/cd` directory on the client side and the `/views_rr` on the server side exist. If you receive a "`permission denied`" message, just export `/views_rr` on the server side.

If your client computer uses a PC operating system such as DOS or NT arbeitet, you can install an NFS client on the PC or install a PC file server such as samba on the server computer. You can obtain samba from the iXOS ftp server **ftp.ixos.de**, or from **samba.anu.edu.au**, under `pub/samba`. The latest version of samba is included with iXOS-JUKEMAN. You can find more information about this package at the samba web site, **http://samba.anu.edu.au**.

Macintosh Clients

iXOS-JUKEMAN allows Macintosh clients to access disks in a jukebox. To export the iXOS-JUKEMAN file system to Macintosh computers, the 'MacFile' service module for Windows NT must be installed. The module can be installed from the 'Network' configuration dialog in the 'Control Panel'. It is either on the Windows NT CD-ROM or on an additional CD-ROM available from Microsoft.

If that service is installed, then a virtual volume can be created with the File Manager (not with the Windows NT Explorer!). Such a volume can be connected from the Macintosh clients over an NT share name.

There is one restriction with the MacFile service: If disks in a jukebox are changed, then the Macintosh clients are not notified about changed or news disks. The above explained volume has to be recreated.

4 Using iXOS-JUKEMAN

4.1 Introduction

This section is devided into taks described for both the GUI and the CLI:

4.2 The server

The jukebox file server is the core of iXOS-JUKEMAN. It controls the devices, and replies to NFS requests and to requests from the iXOS-JUKEMAN local file system for NT. The server and all the programs and files it needs are in the JUKEMAN directory. All paths described here are relative to the JUKEMAN directory.

The server `cdnfsd` runs as a daemon and receives requests from NFS clients, from `cdadm` and from the iXOS-JUKEMAN native file system for NT. It creates `logfile.txt` for messages and a database `volumes` for disk names and properties. It needs a file `server.lic` containing the license key (to use the IFS it also needs a valid `writer.lic`) and a file `server.cfg` containing the server configuration.

To enable task distribution and efficient service to a large number of clients, the server splits into separate threads. The number of threads increases with the number of devices. They share text and data to minimize load on the computer.

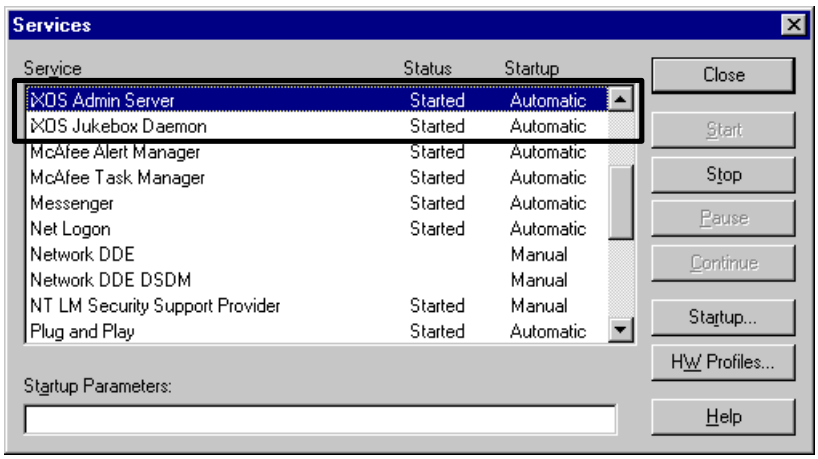
The server controls devices for handling disks and exports the disks in a single large file system in which each disk is represented by a directory. NFS clients need just a single mount, and PC clients connect a single network drive. The server hides the physical positions of the disks. Each disk is represented by a subdirectory, whether it is in a storage slot or in a drive. Clients don’t know if a disk is actually stored in a jukebox, a tower or is a copy stored in another jukebox. They experience faster access to the disk because the server chooses the jukebox with less load.

4.3 Starting iXOS-JUKEMAN

4.3.1 Windows NT:

Starting the server

On NT the services “iXOS Admin Server” and “iXOS Jukebox Daemon” will be started automatically when the server is installed followed by a re-boot of the host. This behavior can be changed with the “Services” table from the “Control Panel”:



Clicking **[START]** and **[STOP]** the selected service can be started and stopped (in that order). Clicking **[STARTUP]** you can select, whether the selected service should be started automatically.

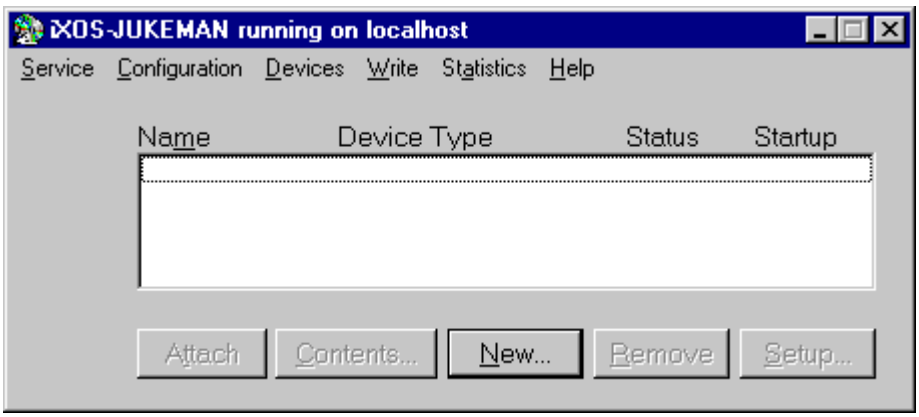
Alternatively, the services can be started from the command line by an administrator or super user with `cdstart.bat` from the JUKEMAN directory or `net start cdnfsd`.

Starting the administration client

You can start the administration client under Windows NT in one of the following ways:

1. **[START]-[PROGRAMS]-[IXOS-JUKEMAN]-JUKEMAN ADMINISTRATION**.
2. Running “`jukeboy`” from the JUKEMAN directory.

NT Administration client



Menu entries	Meaning	Page
Service Configuration De		
Stop Jukebox Service	Start or stop the jukebox service	
Select Host...	Select a host for network administration	96
License Keys...	Enter license keys	34
Exit	Exit the administration client	
Configuration Devices W		
File System Views...	Set up views	72
Volumes...	Edit the `volumes` database	112
Parameters...	Edit server parameters	130
Buffers and Caches...	Set up buffers and caches	42
Devices Write		
Attach	Attach a device to the server	98
Detach	Detach a device from the server	100
Contents...	Show contents, manage disks	112
New...	Set up new devices	60
Remove	Remove devices	92
Setup...	Set up selected device	60
Write Statistics Help		
Single Track at Once...	Burn disks	147
Statistics Help		
Volume Statistics...	Display volume statistics	105
Device Statistics...	Display device statistics	106
CD Statistics per Device...	Display statistics for selected device	107
Reset Statistics	Reset Statistics	
Help		
Help...	Open the online help	
About iXOS-JUKEMAN...	Information about iXOS-JUKEMAN (version)	

The iXOS-JUKEMAN administration client is a graphical user interface (GUI) that can be run on any Windows NT or Windows 95 computer in the network. The network address of the host where iXOS-JUKEMAN is running is stored in the file "`jukeboy.ini`" in the `%systemroot%` directory.

You can configure devices and views on the file system and insert and remove disks from a jukebox using the GUI.

To maintain the configuration, the administration client reads and writes the server configuration file (`server.cfg`) and a device description file for each device.

You can also edit these files manually (see the sections entitled CLI), but is is easier to use the GUI to configure devices and exported views.

The main dialog has a menu and a list of devices. These devices can be 'attached' or 'detached'. 'Attached' indicates that the device is controlled by iXOS-JUKEMAN.

The property 'Startup' indicates whether a device should be attached automatically at the start-up of iXOS-JUKEMAN or manually ('manual'). Using the 'Attach' button, you can add devices from the list to be controlled by iXOS-JUKEMAN.

The buttons can only be selected if you configured devices. If you just started iXOS-JUKEMAN in its virgin state for the very first time devices must be configured as described in "Set up devices" on page 48.

Communication between the GUI and the server is always initiated by the GUI, not the other way round. This is why the server cannot tell the GUI that a device has been detached using an `cdadm` command from the command line or that a device got switched off. The relevant device will still be listed as being attached in the device list of the GUI. If you are uncertain of the current state of the server, either restart the GUI or click the Attach/Detach button. This also applies to situations where the GUI issues a time-out error.

Note:

If you do not use the GUI to configure the devices, you can name the device description and save files anything you want. Put them into the directory where `cdnfsd.exe` resides, since this is where iXOS-JUKEMAN looks for them. However, random names can lead to problems if you use the GUI afterwards, since the GUI only accepts device description files with the name of the device (as stored in `server.cfg`) and the extension `.dev`.

4.3.2 UNIX:

In the JUKEMAN directory, call:

```
./cdnfsd
```

You must be logged in as root. iXOS-JUKEMAN should display a message like this:

```
Starting iXOS-JUKEMAN Version 2.2 Server, build 1007.12
Copyright 1991-1997 iXOS Software AG
... iXOS-JUKEMAN started
```

The server starts up and exports the file systems. The exported file systems (e. g. `/views_rr`) can be mounted with NFS (even locally). The disks do not appear in the directories representing the views (e. g. `/views_rr` or `/views_pc`); these directories' destiny is to be mount points for `rpc.mountd`. These directories must be mounted to access the disks.

4.4 Modifying the set-up

All settings described in the chapter "Setting up iXOS-JUKEMAN" on page 29 (license keys, buffers and caches, devices, views) are not the subject of frequent changes. However, the set-up sometimes needs to be changed, e. g., if new devices should be controlled by iXOS-JUKEMAN. The following sections tell you what to do in these cases.

4.4.1 Change buffers and caches

Note: If you change the properties of a cache (directory cache, data cache) or of the IFS buffer all stored data of the relevant cache or buffer will be lost. All data in the IFS buffer must first be written to disk using the "`cdadm writer flush`…" command (see "Burning disks incrementally" on page 151).

Follow the instructions in "Set up caches and buffers" on page 36.

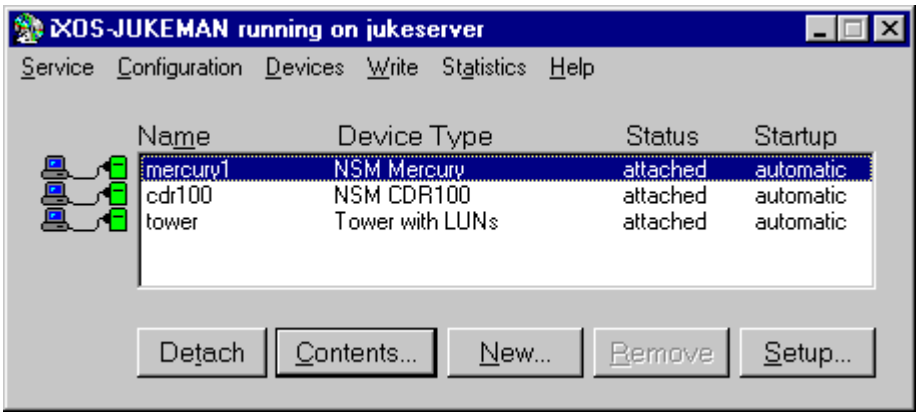
4.4.2 Add new devices

You can set up new devices at any time. Please note that the limit of licensed disks may be exceeded when you attach new devices to the server (see “Set up license keys” on page 31). Disks exceeding the limit appear as “`- limit -`” in the **[CONTENTS]** dialog or with “`cdadm survey`…”. These disks are not visible in the file system.

Follow the instructions in “How to set up devices” on page 60. The section “Attach devices” on page 98 tells you how to attach these newly defined devices to the server.

4.4.3 Remove devices

GUI **Windows NT**



1. Select the corresponding device from the device list.
2. Click **[DETACH]** to detach the selected device.
3. Click **[REMOVE]** to remove the selected device. The device description file and the save file will be deleted. The names of the disks located in the removed device will remain in the `volumes` database of iXOS-JUKEMAN. The advantage is that any renamings will not be lost if the disks are made available again. If you want to delete the disk names from the `volumes` database, read the section “Delete unavailable disks from the database” on page 116.

CLI **UNIX, Windows NT**

1. Change to the JUKEMAN directory.
2. Enter "`cdadm detach` *<device>*" to detach the device. *<device>* is the name of the device to be removed.
3. Remove the corresponding device from the file `server.cfg` (see "Configuration file server.cfg" on page 213). To to this, open the file `server.cfg`. All devices are defined in the `devices` section.

For instance, with the following device configuration,

```
devices {
      list    { p18 mercury }
      p18     { startup { automatic } }
      mercury { startup { manual } }
}
```

to remove the device `mercury` change the section in the following way:

```
devices {
      list    { p18 }
      p18     { startup { automatic } }
}
```

The names of the disks located in the removed device will remain in the `volumes` database of iXOS-JUKEMAN. The advantage is that any re-namings will not be lost if the disks are made available again. If you want to delete the disk names from the `volumes` database, read the section "Delete unavailable disks from the database" on page 119.

4.4.4 Change views

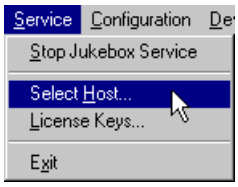
Follow the instruction in “How to set up views” on page 72.

If you change the views manually in the file `server.cfg` the changes can be made active with the command `cdadm cvtree` (see page 217) without having to stop the server. On UNIX, the corresponding directories must be created and exported as described in “Intregrate iXOS-JUKEMAN into the network” on page 77.

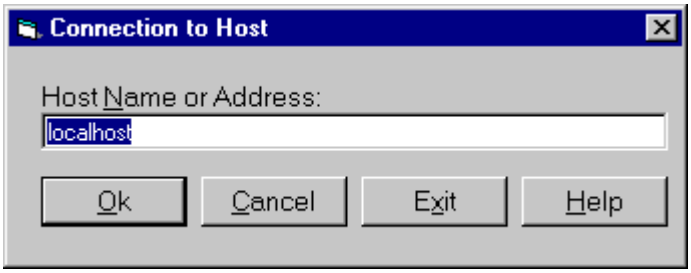
4.5 Network administration

GUI **Windows NT**

The GUI communicates via TCP/IP (port 4072) with the Admin service, which must be running on the (NT) host where iXOS-JUKEMAN is running. In most cases, this will be the local host (the default), but since the GUI can run on any computer in the network, you can choose a particular host.



Select **[SERVICE]-SELECT HOST**:



1. Enter the host name or IP address in the text field of the dialog. If iXOS-JUKEMAN and the GUI run on the same host you can simply enter `localhost`.
2. Click **[OK]** to administer iXOS-JUKEMAN remotely. The administration client tries to contact the administration server on the specified host. If the connection fails you will get the following error message:

If this happens, check the host name and whether iXOS-JUKEMAN is started on the host.

Click **[CANCEL]** to cancel the dialog. Click **[EXIT]** to quit the administration client.

CLI **UNIX, Windows NT**

You can call the administration client `cdadm` with an optional parameter "`-h` *<hostname>*". If this parameter is specified the command will not be executed locally, but on the specified host.

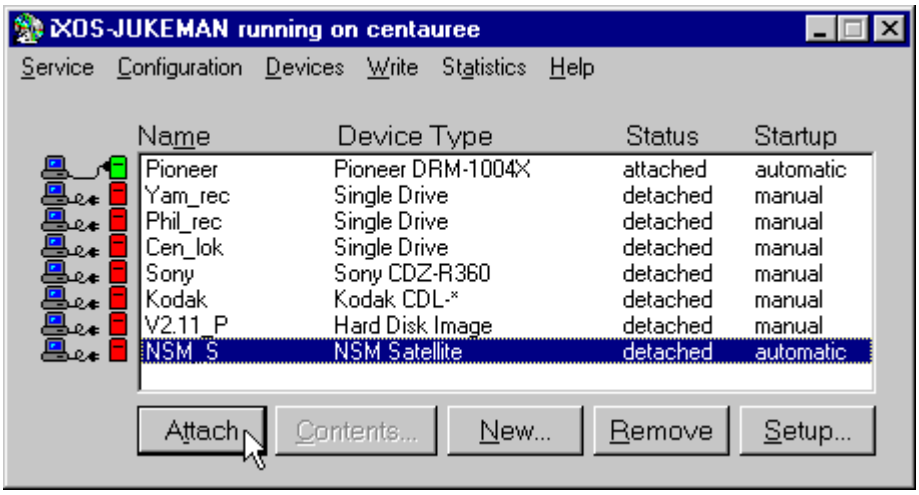
Example:

```
cdadm -h jukeserver attach tower.dev
```

will attach the device with the device description file `tower.dev` to the host "jukeserver".

4.6 Attach devices

GUI **Windows NT**



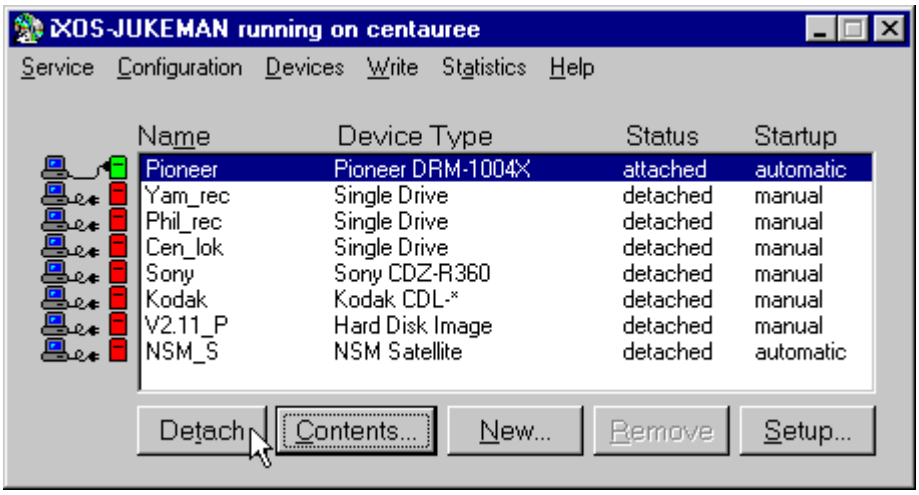
1. Select the corresponding device from the device list.
2. Click **[ATTACH]** to attach the device to the server.

CLI **UNIX, Windows NT**

1. Change to the JUKEMAN directory.
2. Enter `cdadm attach` <*device*>. <*device*> is the name of the device description file. The extension `.dev` may be omitted.

4.7 Detach devices

GUI **Windows NT**



1. Select the corresponding device from the device list.
2. Click **[DETACH]** to detach the selected device from the server.

CLI **UNIX, Windows NT**

1. Change to the JUKEMAN directory.
2. Enter `cdadm detach` <*device*>. <*device*> is the name of the device description file. The extension `.dev` may be omitted.

4.8 Attach devices automatically

Using the command `cdadm` (see section “Command line index” on page 213), devices can be attached or detached dynamically while the server is running. In addition, you may specify a list of devices to be attached automatically when the server starts up.

The server will not respond to file system requests while devices are being attached automatically. If, in contrary, the devices are being attached with `cdadm` after the server start-up, there is a small time gap in which the server will accept file system requests, but will respond with error messages, since the disks cannot be found. This can be avoided if you list the devices in the file `server.cfg`.

This is especially useful with NFS. If the server fails the clients will not be affected. They do not receive error messages, they just wait for the server to reply. This demonstrates the power of the stateless NFS concept.

Format of the devices section in the file server.cfg

The section `devices` contains a subsection called `list`. This is where devices are listed (to be more specific, the section lists the names of the device description files. The extension `.dev` may be omitted). The next subsections specify whether or not each device is to be attached automatically at server start-up. A device for which manual attach is defined has no effect in `server.cfg` (apart from making it known to the GUI in NT, so you can attach it manually through the GUI). This allows you to disable automatic attach of a device without deleting it in `server.cfg`. A device for which automatic attach is defined is attached by the server upon start-up before the server accepts any file system requests. This avoids a time gap in which the file system is present but incomplete.

Example for a device list

Suppose you have two jukeboxes, whose device description files are named `mercury.dev` and `pioneer.dev`, and you want the Mercury to be attached automatically upon server start-up. The `devices` section in `server.cfg` should then look like this:

```
devices {
     list { mercury pioneer }
     mercury { startup { automatic } }
     pioneer { startup { manual } }
}
```

4.9 Display statistics

iXOS-JUKEMAN allows to print statistics about device and disk accesses. There are three types of statistics:

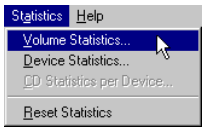
- the amount of transferred data and disk moves concerning the disks in all controlled devices
- the amount of transferred data and disk moves concerning the disks in a specific device
- The amount of transferred data and disk moves concerning the disks in one device (summed up)

The values are with regard to the running server process. To view the statistics for a specific disk, sort the list by disk names. One of the following name formats may be selected: PC format, Rock Ridge format, High Sierra format, and original disk name.

To find out which disks are accessed most frequently, the list can be sorted appropriately. 'Reads' is the number of read accesses to data blocks of 64kB maximum.

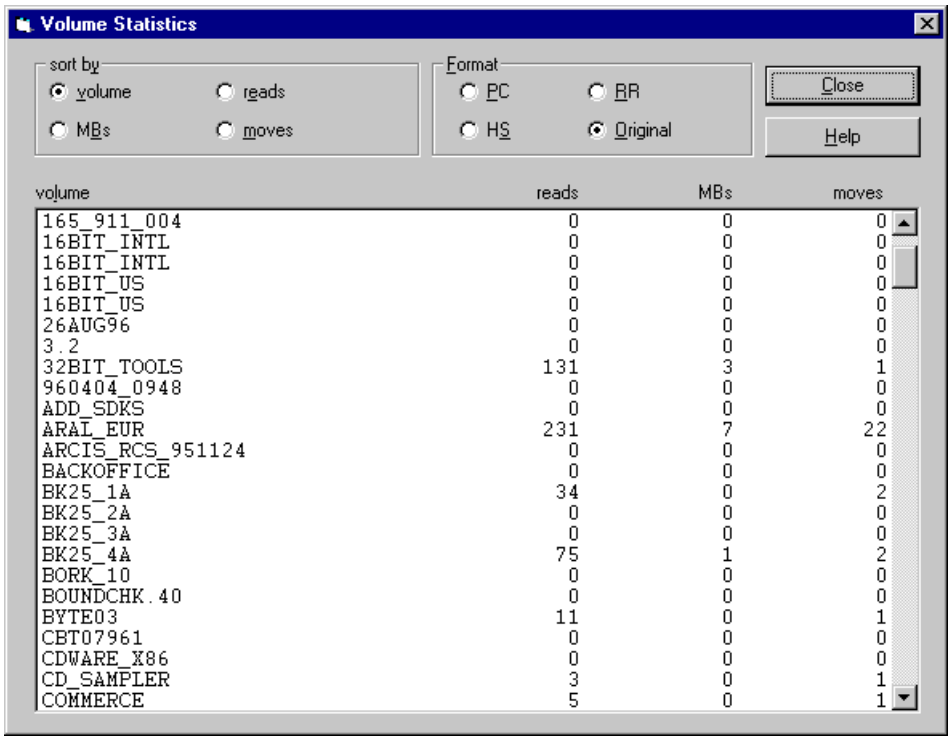
The value 'moves' is the number of moves of a disk into a drive of a jukebox. If this number is very large the overall performance may be improved by inserting the disk into a separate CD-ROM drive or tower.

GUI **Windows NT**



[STATISTICS]-VOLUME STATISTICS:

The following dialog will pop up:

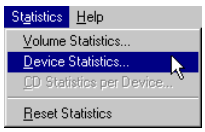


The dialog displays a list of all disks controlled by iXOS-JUKEMAN. Displayed are:

- disk name (**volume**)
- number of read accesses (**reads**)
- MBs read (**MBs**)
- number of moves of a disk into a drive (**moves**)

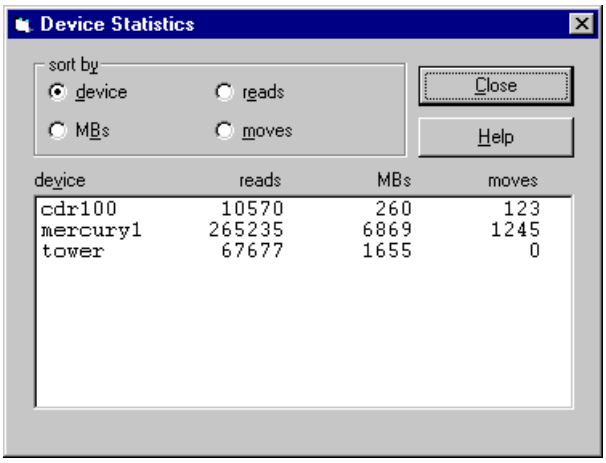
The list can be sorted by any of the four columns (radio button **sort by**).

The disk name can be displayed in original format, PC format, Rock Ridge format, or High Sierra format, depending on the radio button **Format**.



[STATISTICS]-DEVICE STATISTICS:

The following dialog will pop up:

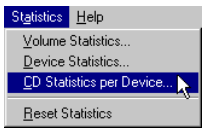


The dialog displays a list of all disks controlled by iXOS-JUKEMAN. Displayed are:

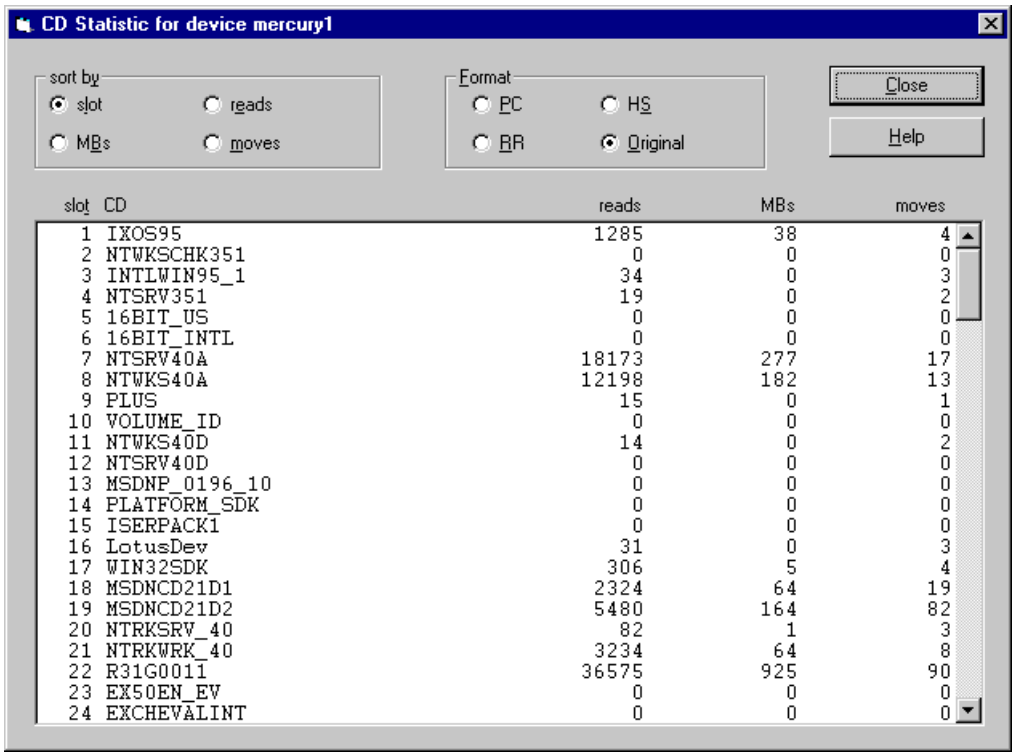
- device name (**device**)
- number of read accesses (**reads**)
- MBs read (**MBs**)
- number of moves of a disk into a drive of the device (**moves**)

The list can be sorted by any of the four columns (radio button **sort by**).

[STATISTICS]-CD STATISTICS PER DEVICE:



The following dialog will pop up:



slot	CD	reads	MBs	moves
1	IXOS95	1285	38	4
2	NTWKSCHK351	0	0	0
3	INTLWIN95_1	34	0	3
4	NTSRV351	19	0	2
5	16BIT_US	0	0	0
6	16BIT_INTL	0	0	0
7	NTSRV40A	18173	277	17
8	NTWKS40A	12198	182	13
9	PLUS	15	0	1
10	VOLUME_ID	0	0	0
11	NTWKS40D	14	0	2
12	NTSRV40D	0	0	0
13	MSDNP_0196_10	0	0	0
14	PLATFORM_SDK	0	0	0
15	ISERPACK1	0	0	0
16	LotusDev	31	0	3
17	WIN32SDK	306	5	4
18	MSDNCD21D1	2324	64	19
19	MSDNCD21D2	5480	164	82
20	NTRKSRV_40	82	1	3
21	NTRKWRK_40	3234	64	8
22	R31G0011	36575	925	90
23	EX50EN_EV	0	0	0
24	EXCHEVALINT	0	0	0

The dialog displays a list of all disks controlled by iXOS-JUKEMAN. Displayed are:

- disk slot (**slot**) and disk name (**CD**)
- number of read accesses (**reads**)
- MBs read (**MBs**)
- number of moves of a disk into a drive (**moves**)

The list can be sorted by any of the four columns (radio button **sort by**).

The disk name can be displayed in original format, PC format, Rock Ridge format, or High Sierra format, depending on the radio button **Format**.

CLI **UNIX, Windows NT**

From the command line statistics can be printed with the command `cdadm survey` (see also page 229).

The order of the parameters is:

1. What should be reported?
2. What should be displayed in the report?
3. Shall the report be contrained (e. g., just one device)?
4. Shall the output be sorted and by what criteria?

The first two parameters are mandatory, the others are optional.

Five list type paramters preceeded with '`-`' are available, one for devices, three for disks and one for drives:

- `-d` print a list of devices (jukeboxes or drives)
- `-v` print a list of disks
- `-n` print the contents of the `volumes` database.
- `-s` like `-v`, but for all slots of a device, even if only a subset of slots is configured to be used (see description of `disks=` in "Slot set-up" on page 55).
- `-r` print a list of drives.

The second parameter, preceeded with '+', determines the source of information. Depending on the list type, only several parameters can be used.

General:

`+d` device names of attached devices (to be more specific: the name of the device description file)

`+n` Total number of slots of a device

`+s` Slot number

`+i` inode number in `volumes` database

Disks:

`+m` type of disk (CD-ROM, rewritable...)

`+R` '`r`', if a recorder is necessary to read, 'a' otherwise

`+a` '@', if the disk is in a drive, '`-`' otherwise

`+u` '+', if the disk can be accessed by the file system, '`-`' otherwise

`+U` time of last access to disk in seconds since 1970

`+S` size of the disk, including free space (in kBytes)

`+I` file system implementation (e. g., `iso`, `hfs`, `ifs`, `ixw`)

`+v` file system-specific information

`+Y` '`rw`', if the disk is writable, '`r`' otherwise

Names:

`+o` original disk name

`+r` disk name in Rock Ridge format (`rr`)

`+p` disk name in PC format (`pc`)

`+h` disk name in High Sierra format (`hs`)

For standard disks the name is printed in the chosen name format. Disks exceeding the limit of licensed disks appear as "`-limit-`". An empty CD-R appears as "`-blank-`" or "`-badCD-`", depending if the recorder can tell the difference. Disks in a format foreign to iXOS-JUKEMAN appear as "`-nostd-`". If it is neither possible to read the disk nor to explicitly judge the disk as being an empty writable disk, it appears as "`-badCD-`". Non-existing slots appear as "`-------`". This can be the case if not all packs are fitted in a jukebox. Finally, empty slots appear as "`-empty-`".

Incremental file system:

`+B` amount of data buffered for a volume

`+W` amount of data written (physically) to the disk

`+w` W+B (total amount of data for a disk)

`+F` S-W (free space on physical disk)

`+f` S-w (free space for further data)

`+T` number of tracks written to a disk

Statistics:

`+D` amount of data (MBytes) read from a disk

`+-D` like `+D`, but set all values to zero afterwards

`+P` number of operations on a disk, i. e. read accesses with max. block size of 64kB

`+-P` like `+P`, but set all values to zero afterwards

`+M` number of movements of a disk into a drive

`+-M` like `+M`, but set all values to zero afterwards

Examples:

The command "`cdadm survey -d +d`" lists the device description file names of all attached devices.

"`cdadm survey -v +dsipr`" prints a list of devices names, slot numbers, inodes and disk names in the `pc` and `rr` format for all disks in all attached devices.

"`cdadm survey -v +oIv`" prints a list of original disk names, file system implementation and file system-specific information, e. g.:

```
Online Docu V6tation hfs BlockSize=512 BlockCount=398748
CSMDN610C           iso BlockSize=2048 BlockCount=320189
asterix             ifs Blocksize=2048 Total=333000 Written=3584
Buffered=1 Used=3585 Blank=329416 Free=329415 Tracks=3
wormimage           ixw Backup=0 BlockSize=1024 FCB=64-65
Data=2048-2048 CTime=878292409 Hid=1053741549 OCTime=878292409
OHid=1053741549
```

Restricting the output:

<*column*>=<*value*> or <*column*>!<*value*>

You can restrict the output using parameters such as d=<*device*> or d!<*device*>. The former prints information for the specified device only.

The latter prints information for all other devices (negative comparison). For a disk list, you can apply restrictions for all columns, even if a column is not selected. Examples: "`cdadm survey -v +sip d=juke.dev`" will print all disks of `juke.dev`. "`cdadm survey -v +dsipr p=ixos96`" will print the disk with the PC format name `ixos96`.

Note: `(t)csh` and `bash` users must preceed `!` with a backslash to avoid its usual function as History operator.

Sorting the output:

Finally, you can specify how the output should be sorted:

`s:`*<criteria>* sorts the output list according to the specified options.

<criteria> is a list of column names (may be preceeded with '`-`' to reverse sort order). The output is sorted by these columns.

Examples:

"`cdadm survey -v +dsipr s:d-S`" lists the disks, sorted by device name and within a device, by slot number. Like restrictions, you can use sort options, even if the specified column is not printed:

Thus "`cdadm survey -v +dspr s:i`" prints all disks sorted by inodes, but does not print any inode numbers. This may be useful if you want to see disks in the order in which they were made known to the server, because the server associates inode numbers with disks sequentially.

The results of commands to show amounts of buffered or written data, free space, and so on, are expressed in kBbytes (1024 Bytes).

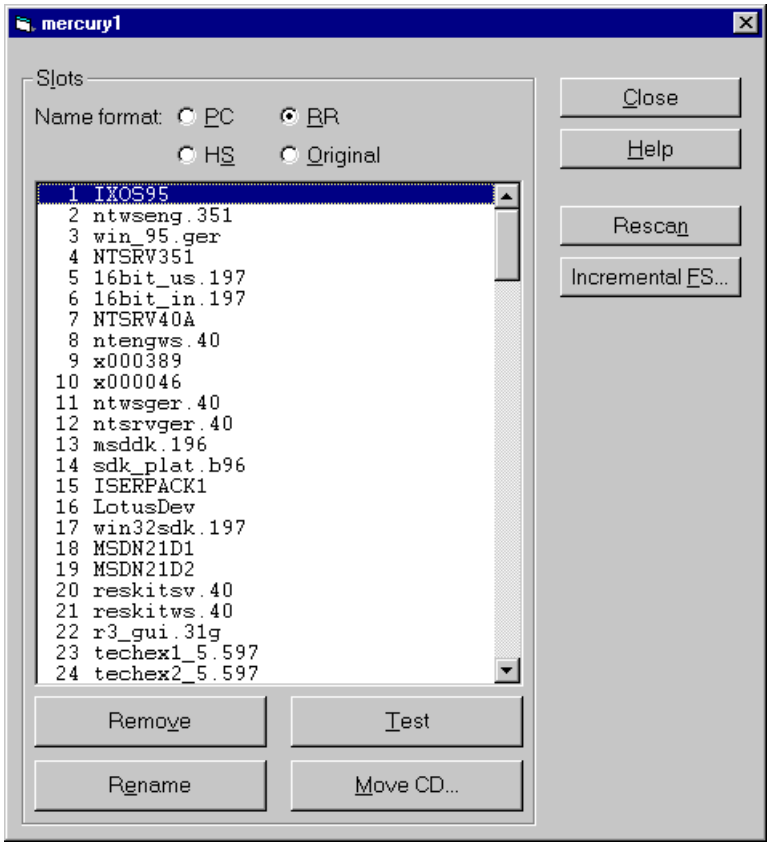
4.10 Manage disks

GUI **Windows NT**

The [CONTENTS] dialog

Select **[CONTENTS]** or **[DEVICES]-CONTENTS**. The dialog will display a list of all slots and their current contents. The disk names appear in the selected name format for standard disks. The following special labels may also appear:

Lable	Meaning
`-limit-`	This label appears if the limit of licensed disks is exceeded. These disks will not be visible in the file system.
`-blank-`	Empty disk. Will appear only if the recorder is able to detect empty disks, otherwise `-badCD-` will appear.
`-badCD-`	Empty or not readable/defect disk. Since some recorders will not be able to detect empty disks, the distinction cannot uniquely be done.
`-empty-`	Empty slot.
`-nostd-`	Disks in a format foreign to iXOS-JUKEMAN.
`-------`	Non-existing slots (e. g. missing packs in a jukebox).
`x?????????`	Generic name (e. g., `x0000327`). A disk will be assigned such a name, if another disk with the same name is already stored in the `volumes` database or if the disk has got no name. These disks can be renamed later.



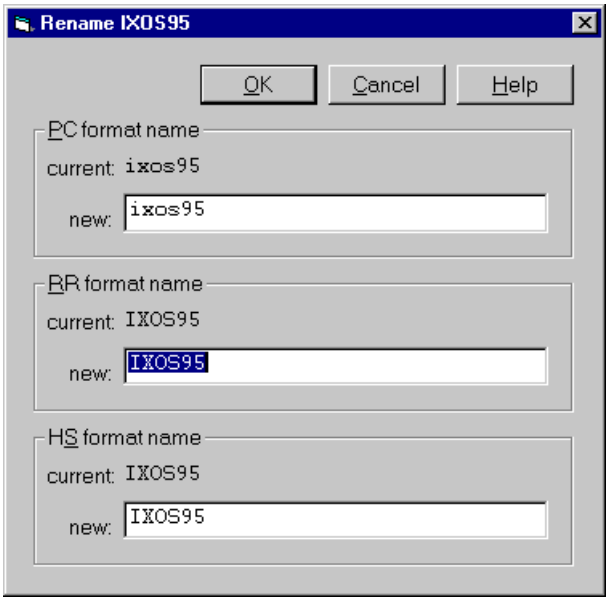
This dialog lists all slots of the currently selected device. The list displays the slot number followed by the disk name in the selected name format. By default, the original name of the disk is displayed (**Original**). The name format can be selected with one of the radion buttons **PC**, **RR**, **HS**, and **Original**.

Rename disks

Note: iXOS-JUKEMAN 2.2 allows disks to be renamed using the operating system functions (e. g., Windows NT Explorer or UNIX `mv` command). The renaming takes place for the name format defined for the view.

The disk name can only be changed if one of the three name formats PC, RR or HS is selected. The original name stored on the disk cannot be changed for obvious reasons. The Rename dialog allows to change the name for any of the three name formats.

1. Click on the disk to be renamed.
2. Click **[RENAME]**. This button can only be clicked if a single disk is selected. The selected name format must not be “Original”. The following dialog will pop up:



3. Enter a new name for the desired name format.
4. Click **[OK]** to accept the changes. Click **[CANCEL]** to discard the changes.

Insert/remove disks

Depending on your type of jukebox (mail slot or not), the GUI offers two different buttons to change disks:

The button **[CHANGE CD]** is available, if the jukebox does not have a mail slot to change disks. To change a disk in this type of jukebox, select the slots in which disks should be changed and click **[CHANGE CD]**. Then you must change the disks using the corresponding mechanism of the jukebox. Afterwards, the server will scan the selected slots and display their contents as if you clicked on **[TEST]**.

If your jukebox has a mail slot (e. g., Mercury, Kubik or standard juke-boxes), the GUI offers the buttons **[INSERT]** and **[REMOVE]** instead of **[CHANGE CD]**. They can be used to change disks using the mail slot.

Click **[INSERT]** to instruct the server to look for an empty slot and open the mail slot. Insert a disk and close the mail slot. If one or more slots are selected, only this range of slots will be scanned for empty slots.

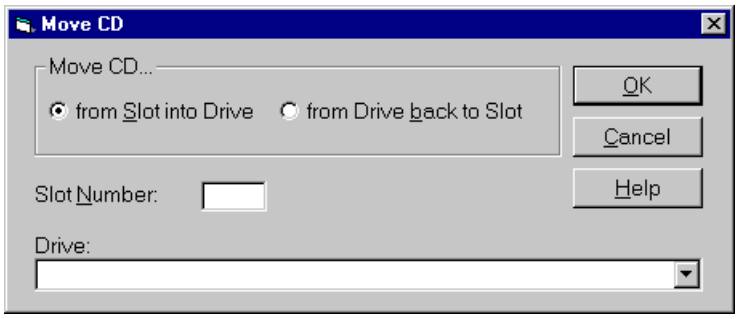
Click **[REMOVE]** to remove a disk from the jukebox. This button is only available if a single occupied slot is selected.

Note: Please note that disks can be changed manually in single drives and towers, without using iXOS-JUKEMAN. These drives

will be scanned periodically for new disks (see server parameter dcheck in section “Server parameters” on page 123).

Move disks to or from a drive

If a disk is accessed by a client request it will be moved to a drive automatically. If you want to move a disk into a specific drive (e. g., a recorder) click **[MOVE CD]** and select the slot number and the drive:



To move a disk from a drive back to its slot select “**from Drive back to Slot**” and select the drive.

Test disks

If you are uncertain about the contents of a slot (e. g., if you changed disks manually without using the GUI) you can explicitly test the contents of one or more slots. To to this, select the corresponding slots and click **[TEST]**.

Rescan jukebox

Click **[RESCAN]** to refresh the internal memory of the jukebox. This is required for some jukeboxes with a non-volatile memory if you changed disks manually, and can do no harm when performed on other jukeboxes.

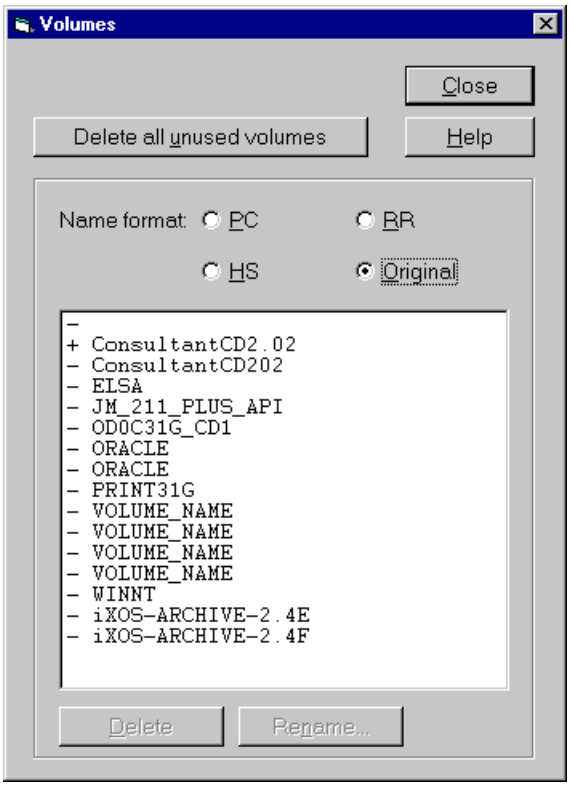
Please note that a complete rescan can take some time for bigger jukeboxes (such as Pioneer DRM-5004x, JVC MC-200/600, DISC CD-CHG DJ-200/600). For these jukeboxes, a partial rescan can be performed if the relevant disks are tested.

[INCREMENTAL FS]

The incremental file system is described in section “Burning disks incrementally” on page 151.

The [VOLUMES] dialog

Select **[CONFIGURATION]-VOLUMES**. This dialog displays the disk names of all disks known to iXOS-JUKEMAN (and the renamings). All disk names scanned by iXOS-JUKEMAN are stored in the internal `volumes` database, even if you remove disks from a jukebox or drive.



Available disks are marked “+”, currently unavailable disks are marked “-”. A disk is unavailable either if the device containing the disk is not attached or if the disk was removed from the device. Clicking **[RENAME]** will open the dialog described in “Rename disks” on page 113.

Delete unavailable disks from the database

Select a disk name from the list and click **[DELETE]**, or press the ‘Delete’ key to delete the name of an unavailable disk from the `volumes` database (the selected name format must not be ‘Original’). You need to confirm

this before the name is actually deleted. To remove the disk names of all unavailable disks, click **[DELETE ALL UNUSED VOLUMES]**.

CLI **UNIX, Windows NT**

List disks

You can display a list of all disks using the following command:

```
cdadm survey -v +idsoprh
```

The following columns are printed (see page 229 for a list of other columns):

(1) Disk ID in the `volumes` database,
(2) name of the device description file,
(3) slot number,
(4) original disk name,
(5) `pc` name,
(6) `rr` name,
(7) `hs` name.

The output will look similar to this:

(1)	(2)	(3)	(4)	(5)	(6)	(7)
362	tower.dev	1	MSVC42	msvc.42	msvc.42	MSVC42
325	tower.dev	2	MSVC41	msvc41	MSVC41	x000325
137	tower.dev	3	OFF95_Z_01	offpro95.eng	OFF95_Z_01	OFF95_Z_01
535	tower.dev	4	VSENT_CD1	vs_stud1.97	vs_stud1.97	VSENT_CD1
0	tower.dev	5	-empty-	-empty-	-empty-	-empty-
541	tower.dev	6	VSENT_CD3	vs_stud3.97	vs_stud3.97	VSENT_CD3

In some cases you will notice strange disk names such as `x000325` in the second row. This is a generic name assigned by iXOS-JUKEMAN. A disk is assigned a generic name if the disk does not have a name or if the disk name is already stored in the `volumes` database for another disk. If a disk is removed from a device its name remains in the database. All renamings for the disk in the `pc`/`rr`/`hs` format will be remembered if the same disk is inserted again at a later time. To remove the name of an unavailable disk from the database see “Delete unavailable disks from the database” on page 119.

Rename disks

Note: iXOS-JUKEMAN 2.2 allows disks to be renamed using the operating system functions (e. g., Windows NT Explorer or UNIX `mv` command). The renaming takes place for the name format defined for the view.

`cdadm rename [[-`*<nf>*`] <old> [<new>]]`

renames the disk from *<old>* to *<new>* for the specified name format *<nf>* which must be either `pc`, `rr` or `hs` (see “Set up views” on page 67). *<old>* must be an existing disk name for *<nf>*, *<new>* must not exist. A disk name exists if it is stored in the `volumes` database even if the disk with the corresponding name is currently not inserted in a device. This is to avoid disk name conflicts when attaching and detaching jukeboxes. The old name will be replace by the new name in the database `volumes`.

Delete unavailable disks from the database

A disk name will remain stored in the `volumes` database if the disk is removed from a device. Its name can be deleted explicitly in the following way:

`cdadm rename -`*<nf>* *<old>*

In comparison to renaming a disk the *<new>* name is missing. To delete the disk names of all unavailable disks from the `volumes` database (i. e., disks listed as “`-`” by `cdadm survey -n +uo`), use the command `cdadm rename` without any parameters. Please note that all renamings for these disks will be lost!

Insert/remove disks

You can use the commands `cdadm insert` and `cdadm remove` to insert disks into or remove disks from a device. The command `cdadm import` is identical with `cdadm insert`, the command `cdadm export` is identical with `cdadm remove`.

Due to the different types of jukeboxes there are four main scenarios for inserting and removing disks (see also the relevant sections in “Supported jukeboxes” on page 163):

i) For **single drives** and drives in **tower jukeboxes** a disk change is very easy: Just insert the disks into or remove them from the drives. No interaction with the server is required. The server will check these drives for disk changes periodically (see parameter `dcheck` in “Server parameters” on page 123). The changed disks do not have to be tested anymore, as with earlier versions of iXOS-JUKEMAN. Please note that for this feature to work on Windows NT, the autorun feature has to be disabled (see page 23).

ii) The jukebox provides a **separate mail slot**.

`cdadm remove <device>`

instructs the server to move a disk from the jukebox to the mail slot. If the jukebox contains a bad disk it is moved to the mail slot preferably. Otherwise the server moves the first found disk to the mail slot. For nearly all jukebox types, the mail slot will be ejected automatically. You must then take out the disk and close the mail slot again. With the Kubik jukebox after the acoustic signal the mail slots can be opened manually. Then you can take out the disk and close the mail slot again.

To insert a disk use the command:

`cdadm insert <device>`

This command instructs the server to search a free slot and then open the tray or release the mail slot. Insert a disk and press the Enter button or close the mail slot. The jukebox tests the disk afterwards.

You can restrict the choice of disks or slots using additional parameters:

`cdadm remove <device> <name>`

instructs the server, to remove a disk called *<name>* in the default name format.

`cdadm insert <device> 20-30`

instructs the server to import a disk into one of the slots 20 through 30. An error message is printed if none of these slots is empty.

Similarly:

`cdadm remove <device> 27`

instructs the server to remove the disk in slot 27.

iii) The pioneer500 has a **virtual mail slot** which is in fact a position of the robot. The grundig35 has a mail slot which is hidden under an additional door. You can use all of the above commands, but with the additional switch `-f` (`cdadm insert -f` ... or `cdadm remove -f` ...). As soon as the command returns, the picker of the pioneer500 is in its change position, or the door of the grundig35 may be opened. Open the jukebox door and insert or remove the disk The time-out for the disk change is specified by the `trayto` parameter (see “Server parameters” on page 123) which is 60 seconds by default. iXOS-JUKEMAN waits for the door to be closed and continues to answer the accumulated file system requests for the jukebox. You do **not** need to issue the following command:

`cdadm testcd` <*device*>

which had to be issued in earlier versions to tell the server the disk change is finished.

iv) Jukeboxes with **no mail slot**, e. g., jukebox types cdr100, sony, pioneer6, and pioneer18. For all these devices, `cdadm insert` and `cdadm remove` request the server to block incoming requests and free the drives. Then you can manually change the disks. After making changes use `cdadm testcd` to tell the server which disks are changed and that the device can resume normal operation. Clients do not receive error messages, they simply think the server was slow for a while. A typical command sequence is:

`cdadm insert` <*device*>	(blocks user requests - now change disks 2 and 3)
`cdadm testcd` <*device*> `2-3`	(server inspects disks and resumes normal operation)

Move disks to or from a drive

`cdadm movecd` <*device*> <*drive*> <*slot*>

Moves a disk from <*slot*> to <*drive*>, where the latter is an integer in the range of 1 to the number of drives in the respective jukebox (e. g., 4 for a Mercury). If the slot is 0 or not specified, the server will move the disk in drive <*drive*> back to its slot.

Example:

`cdadm movecd jb.dev 3 42`

moves the disk from slot 42 to drive 3.

`cdadm movecd jb.dev 3`

moves the disk from drive 3 back to its slot.

Note that the drive must be active (not disabled by a `drive=!` line in the device description file) to be accessible for move commands (see “Drive set-up” on page 54).

Test disks

`cdadm testcd` <*device*> <*list*>

The command `testcd` can be used to test the contents of slots, especially to make newly inserted disks known to the server.

Example:

`cdadm testcd jb.dev 27`

instructs iXOS-JUKEMAN, to move the disk from slot 27 to a drive and to check its contents.

`cdadm testcd jb.dev 12-45`

tests the disks in slots 12 through 45. See also “`cdadm testcd <device> <list>`” on page 233.

4.11 Server parameters

You can change the server behavior using several parameters, though the default values are usually quite sensible. `server.cfg` override default values and command options override default values and `server.cfg` values.

A few parameters can be changed dynamically while the server is running. See “Change server parameters” on page 130.

4.11.1 Overview of server parameters

Dynamic parameters

`autodc=1`

This parameter controls the caching of the directory structure of a disk (see “The directory cache” on page 36). If a disk is inserted into a drive, iXOS-JUKEMAN checks if the root directory. If it is not in the cache then `autodc` controls the further behavior. If set to 0, no caching is performed. If set to 1, caching is performed only if a permanet hard disk directory cache is configured. When set to 2, iXOS-JUKEMAN will cache the entire directory structure, even if only a RAM cache is configured. The dynamic caching performed due to file system requests is not affected by this parameter.

`blanks=0`

After changing a disk or issuing a ‘test’ command iXOS-JUKEMAN tries to determine the type of disk. For empty CD-Rs or disks which are not ISO 9660-conforming or non-finalized disks written with the incremental file system (see “Burning disks incrementally” on page 151) this test cannot be completed successfully with many reader driver. In these cases the disk will be tested in a recorder drive to determine the disk type.

The parameter `blanks` can be used to speed up the test procedure. If set to 0, disks are first tested in reader drives (faster than recorders), and eventually in recorders a second time. If set to 1, disks are always tested in recorder drives, if available. If set to 2, no testing is performed and the disks are assumed to be blank.

`fullvn=0`

By default, disk names in the PC name format are converted to 8.3 in lower case. If `fullvn` is set to 1, only the conversion to lower case is done with up to 32 characters. This is useful for networks, where only Windows NT or Windows 95 PCs are used, but not if Windows for Workgroups clients are used.

`hfsiso=1`

iXOS-JUKEMAN supports Apple HFS CDs and Hybrid CDs which contain both HFS and ISO 9600 directory structures. For Hybrid CDs, by default iXOS-JUKEMAN reads the ISO 9660 directory structure and ignores the HFS structure. If this parameter is 0, preferences are reversed, and Hybrid CDs will appear as HFS, not as ISO 9660..

`ignore=0`

All other values than 0 cause the server to ignore all file system requests. This can be changed dynamically, for example to block the server for awhile.

`iotimo=60`

Time-out in seconds until failed disk reads are abandoned.

`loglev=4`

iXOS-JUKEMAN maintains a log file called `logfile.txt` in the JUKEMAN directory. The messages in this file are classified in ten log levels, where level 0 is of highest priority (see also “Log file logfile.txt” on page 249). The parameter `loglev` is the limit that keeps log messages with a higher log level than this limit out of the log file. For your first try, use a value of 5 or 6 to see if everything works fine.

`lwords=5`

The iXOS-JUKEMAN server internally stores last recent log messages in a buffer. If problems occur, than this buffer is written to the log file. This parameter sets the log level for the ‘last words’.

`mdelay=3`

iXOS-JUKEMAN has a sophisticated adaptive scheduling policy for accessing disks. It learns from accesses in the past. When the server has to move a disk out of a drive to insert a new one, the server must first select, which disk will be removed from the drive to make room for the new disk.

It is often preferable to delay this disk exchange as data requests are usually queued for the current disk. This parameter sets a limit according the delay. Value 0 means a strict serial use of the access queue (not very sensible). The greater the value the longer the delay even if there is not access to the disk.

`rahead=3`

Specifies the number of chunks the data cache tries to read ahead.

`reject=1`

Specifies how incomplete disks are treated (e. g., if the burning process for a CD was interrupted). For `reject=0`, iXOS-JUKEMAN tries to read as much as it can from the incomplete disk. For `reject=1`, no incomplete iXOS-JUKEMAN-CDs will be accepted. If set to 2, no incomplete disks at all will be accepted.

The default is `reject=1`. If the disk is not fully readable, it is very likely that some error occurred during burning the disk, and it appears as a bad disk. This setting will make sure to mark only those disks as readable where disk length and actually written data match.

`trayto=60`

This parameter can be used together with Kodak CDL 144, Hyundai HAS-550, Plasmon D-Series, and Sony CDL-2*** jukeboxes. It is the time in seconds that the server waits until the mail slot of a jukebox is closed manually. After that time, the mail slot tray is closed automatically and the new (or old) disk is tested.

For the Pioneer DRM 5004 X and the Grundig GMS 1035 the parameter has a special meaning due to the virtual mail slot and the door hiding the mail slot, respectively: It specifies the time-out for the user to open the door, exchange the disk, and close the door again for a disk change. After this time-out normal operation is resumed.

For all other types of jukeboxes this parameter is irrelevant.

The following table lists the dynamic parameters as well as the units, the default values, the minimum and the maximum values:

Table 5 - Dynamic parameters

Name	Unit	Default value	Minimum	Maximum

Name	Unit	Default value	Minimum	Maximum
autodc		1	0	2
blanks		0	0	2
fullvn		0	0	1
ignore		0	0	1
iotimo	seconds	60	0	3,600
loglev		4	0	9
lwords		5	0	9
mdelay		3	0	99
rahead		3	0	1000
trayto	seconds	60	0	99,999,999

Static parameters

`cdnfsp=100003`

This parameter defines with which program number iXOS-JUKEMAN registers its NFS service, if no other program of that kind exists. Value 0 means no registration.

`dcheck=300`

iXOS-JUKEMAN performs a periodic disk check for single drives and towers. This simplifies changing of disks, since no interaction with the server is required. The change is detected automatically.

The parameter defines a time intervall in centiseconds. After this time iXOS-JUKEMAN checks whether a disk has been changed. Value 0 turns off the check.

`jobnum=192`

Defines how many file system requests the server can queue internally. If you have more than 1,000 clients, increase this number.

`maxcvt=1000`

This parameter defines how many nodes the view tree of the controlled disks can have.

`maxthr=40`

Specifies how many processes can be started.

`mountp=20000234`

Defines the program number the server uses to register its mount service if there is no such program already registered. Use 0 for no registration at all.

`nonfsd=0`

Set this parameter to 1 to prevent the NFS server built in iXOS-JUKEMAN from starting. This is only useful on Windows NT, as NFS is the only way to connect to the file system if iXOS-JUKEMAN is running on UNIX.

`portno=4027` (UNIX) or `2049` (NT)

Defines the UDP port for NFS and `cdadm` requests.

`rtrack=131072` (128 KB)

Defines the size of chunks of files to be cached in the data cache (in RAM or as a file). This value must be a multiple of 8,192 (8 KB) sein.

`synclm=0`

With a value of 1 log messages will not be buffered but written directly to the log file.

`waitpm=0`

This parameter defines the time (in seconds) the server waits for a delayed portmapper to be started.

The following table lists the static paramters as well as the units, the default values, the minimum and the maximum values:

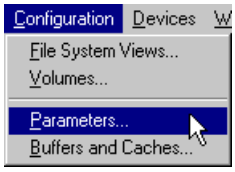
Table 6 - Static parameters

Name	Unit	Default value	Minimum	Maximum
`cdnfsp`		100,003	0	99,999,999
`dcheck`	1/100 seconds	300	0	999,999
`jobnum`		192	9	8,192

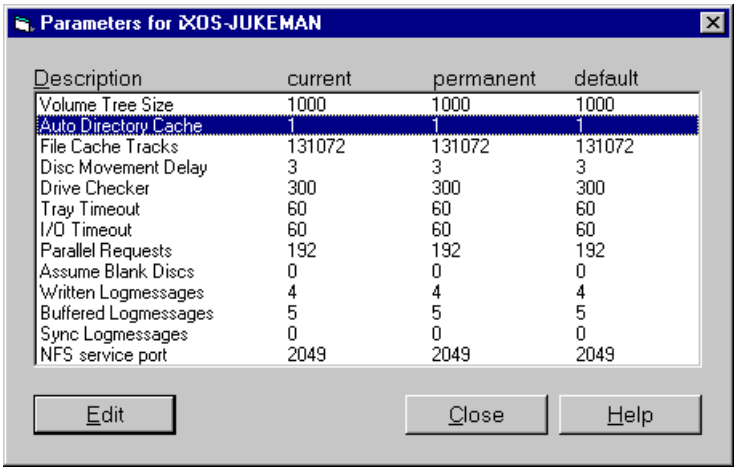
Name	Unit	Default value	Minimum	Maximum
maxcvt		1,000	10	65,536
maxthr		40	12	1,024
mountp		20,000,234	0	99,999,999
nonfsd		0	0	1
portno		4,027 (2,049)	1	65,536
rtrack	bytes	131,072	8,192	16,777,216
synclm		0	0	1
waitpm	seconds	0	0	3,600

4.11.2 Change server parameters

GUI **Windows NT**

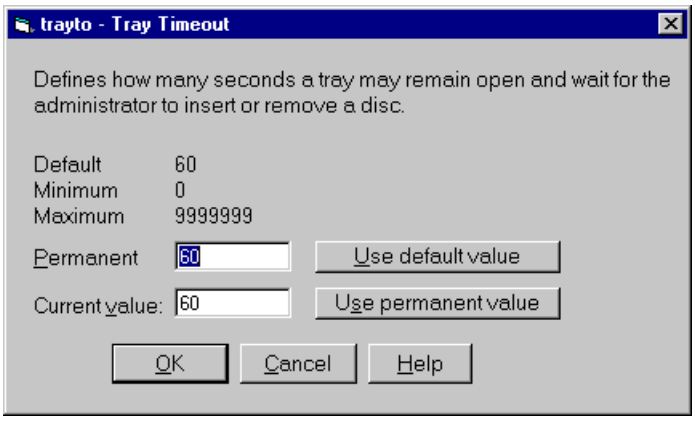


Select **[CONFIGURATION]-PARAMETERS**:



The dialog shows a list of server parameters along with their **current** values, their **permanent** values from the file `server.cfg` and the **default** values.

Select the parameter you wish to change and click **[EDIT]**, double-click the parameter.



In the "**Permanent**" text field, the value in the file `server.cfg` can be set. In the "**Current value**" text field, the value can be changed dynamically while the server is running. This only works for some parameters (see "Table 5" on page 125).

Click **[OK]** to confirm the changes, click **[CANCEL]** to discard the changes.

CLI **UNIX, Windows NT**

In `server.cfg`, all server parameters have the common format <*key*> { <*value*> }. <*key*> is the name of the parameter and <*value*> is a non-negative integer in decimal or hex notation.

Change parameters permanently

To change a value permanently, add an entry in `server.cfg`. If there is no `parameters` section, create it and list all parameters to be changed. For example, you want to change the log level of the server which is set with the parameter `loglev`. The default value is 4, and you want to use log level 5. Enter the following section in your `server.cfg` file:

```
parameters {
     loglev { 5 }
}
```

Change parameters at server start-up

If you want to change the log level (or any other parameter) for a single server session only, start the server in the following way:

```
./cdnfsd loglev=5
```

The server will start with the log level set to 5.

Change parameters dynamically

If you want to change a server parameter while the server is running, type (for instance):

```
cdadm setpar loglev 5
```

and the parameter will be changed dynamically. Please note that this method works for just a few parameters (see “Table 5” on page 125).

4.12 Burning disks/writing incrementally

It is possible to write data to disks in two ways with iXOS-JUKEMAN, namely, in batch mode or incrementally and transparently using the file system.

Batch mode was also a feature of version 1.3b of iXOS-JUKEMAN. The writeable disks are placed in the jukebox and the recorder is reserved for the write process. Once JUKEMAN is instructed to insert a disk into the appropriate drive, the burning process begins in the same way as a disk would be burned in a normal stand-alone drive. Finally the disk is returned to its original slot. It is then possible to include this disk into the overall group of disks, which are an already visible and accessible part of the file system, by clicking on Test (c.f. “Manage disks” on page 112). In this way you achieve a visible migration of data: First the data is collected on a hard disk, before it is written to the disk to be burned and finally integrated into the file system. The drive used for burning is reserved by deactivating same in the device description file (see also “Drive set-up” on page 54). It is also possible to reserve the required drive dynamically during normal operation of the jukebox (e.g to reserve drive 1, simply use the command `cdadm detach jb.dev -d 1`). It is then possible to move a disk to this drive with the command `cdadm movecd jb.dev 1 <slot>` or remove a disk from this drive with the command `cdadm movecd jb.dev 1`. When the drive no longer needs to be reserved, it can be used again for reading with the command `cdadm attach jb.dev -d 1`.

Write process	**Description**	**GUI**	**CLI**
		see page:	
Single Track at Once	133	147	142
Incremental file system	151	156	158
WORM file system	160	—	162

4.12.1 Burning disks (Single Track at Once)

The writing software in iXOS-JUKEMAN can transport the data to be written to a CD, PD, WORM or MO recorder drive. To actually burn a disk, you need the writing software `cdglow` together with `iso9660`, the latter used to generate the required standard ISO 9660 file system. Windows NT also supports the burning of disks from the GUI.

cdglow

Please read the following before attempting to burn a disk:

Burning disks requires a steady stream of data at a constant rate. CD recorders which operate at the basic speed require a data transfer rate of 150 kB per second. With dual- and quad-speed recorders, the required data transfer speed is 300 and 600 kB per second respectively. If the data available in the internal buffer of the CD recorder is not sufficient to maintain this rate, then the burning process is prematurely interrupted with the SCSI error message: "`buffer underrun`" and the CD can no longer be used.

iXOS-JUKEMAN is designed to fully exploit the features of the controlled devices. However, it cannot compensate for an insufficent hardware set-up or operating errors.

The following errors typically cause such buffer underrun errors:

- If the data source is not directly connected to the jukebox, i.e. if the data is being transferred over a network.
- If you are copying data from a CD drive, whose speed is the same or less than that of the recorder drive. An example of this is if you copy data from a quad-speed to the Brenner Yamaha CDE 100 II with variable speed capability. In this case however, it is possible to reduce the effective recording speed with the `cdglow` options `-f1` and `-f2`.

Another cause of error is when there are too many devices attached to the same SCSI controller as that used by the CD recorder. Some recorders cannot even operate if there are other devices attached to the same controller. In this case it is required to use separate controllers for the read drives and the recorder drive. In some cases, however, this problem can be avoided by giving the recorder drive a higher priority (e.g. 6) than the other drives.

In addition to this, no activites or real-time processes, which would overload the hard disk or SCSI busses, should be allowed to run while burning a CD.

If you are unsure about the properties of a device, the `cdglow` command should be run with the option `-p`. Though it is usually not possible to simulate the laser performance, this option allows the recording process to be simulated and whether buffer underrun error is likely to occur. If this is the case, either the buffer can be enlarged or the speed of the CD recorder can be reduced with the options `-f2` or `-f1`. If you plan to copy a CD, it is also possible to copy the contents first to the hard disk to the ISO 9660 standard. The write command will then use this file as the source.

The actual write command is as follows:

```
cdglow [-p] [-v] [-w] [-c] [-f1|-f2] [-b size]
       [-s source] [-t target] [-l length] [-a size]
```

With this command, the data is is written to the CD (WORM or MO) in ISO 9660 format on a single track.

`-p`	preview mode: data is received but not burned
`-v`	on completion, verify that there are no differences between the source data and what actually now appears on the CD.
`-w`	write to WORM and MO.
`-c`	only check the size of the source.
`-f1`	single-speed recording-the default is the highest possible speed.
`-f2`	dual-speed recording
`-b` *<size>*	size of the RAM buffer (in decimal or hexidecimal) –default is 8 MB.
`-s` *<source>*	*<source>* contains the ISO 9660-file system - default is `stdin`.
`-S`	indicates that the source is not a CD drive.
`-t` *<target>*	*<target>* is the CD recorder - default is `stdout`.
`-T`	indicates that the target is not a CD recorder. This mode allows up to 128 KB to be written to the hard disk without a licence.
`-l` *<length>*	indicates how many bytes should be written.- default is the whole ISO image.

`-a` *<size>* the granularity of the control-output, which determines the frequency with which confirmation messages are output when recording data - default value is 1 MB, which corresponds to `-a 0x100000`. This means that for every megabyte of data written to the CD, a message will be output. The actual value provided for *<size>* will be rounded up to the nearest MB. If *<size>* is set to 0, then no messages are output.

Successful completion of the `cdglow` command is confirmed by the return code 0: other values indicate that an error occurred. Such errors are protocolled in `stderr`.

When working with UNIX, the `cdglow` command must either be run under `root` or have its s-(„sticky“-) bit set. Otherwise the speed of the data stream is not properly maintained. The equivalent user under NT is Administrator.

A variety of data sources can be used for `cdglow`, an example of which is a file or partition with a master image formatted to ISO 9660, or another CD drive. In the latter case, it is possible to copy the contents of one CD onto another, providing the speed of the recorder is slower than that of the source.

A more advanced application is, while producing the ISO 9660 file system, to pipe this file system simultaneously to `cdglow`. The software is optimized, so that the real-time capabilities of the operating system are fully exploited to maintain the constant rate of data transfer. This is also the case when the source data is dynamically generated. The precautions mentioned above must still be observed, however.

It is important to be aware that certain drives in the IMS range simulate (“`-p`”) the burn process so accurately, that successful completion of such a simulation requires opening and closing the mail slot, or removing and reinserting the caddy. This is necessary to simulate the actual removal of the CD during a genuine recording. During an actual genuine burning of a CD, the door of the drive should of course not be opened. This is also the case for a simulated recording process with the above family of drives and to prevent the door being opened by mistake, the relevant functionality is deactivated during a simulation. However, should the simulation not be successful, it is sometimes necessary to turn the recorder drive off and on again in order to be able to remove the disk. CD recorders in this range include a variety of drives from Grundig, Hewlett Packard, Kodak, Matsushita, Philips, Pioneer, Plasmon and Yamaha.

iso9660

`iso9660` is program which formats a file system to the ISO 9660 standard. This allows the use of the Rock Ridge extensions, which contain information about UNIX file names and permissions. A file system resulting from this program can be written to its own partition or file before it is burned onto disk. Alternatively, depending on the hardware (see above), it can be channelled directly to `cdglow`, whereupon the need for such a large interim storage is avoided. In either scenario changes to the source file system should be avoided while `iso9660` is running and ensure that the target file system does not share the same directory path of another image produced by an earlier running of the program.

Typically, a Level 1 ISO 9660 file system is generated, in which the file names are converted to conform to the standard 8.3 convention. A limitation of this convention is that only capital letters, numbers and underscores can form such a file name. This is however necessary to provide a file system which is almost universally accessible and readable.

`iso9660` is not limited to this convention, however, and with appropriate parameters certain extensions are permissable. This will of course have the effect, that not all operating systems will be able to read the files, especially if iXOS-JUKEMAN is not installed on the operating system in question. For this reason, it is advisable to carry out tests before burning CD's whose file systems exploit this feature.

ISO 9660-conforming parameters and options

`rr`

This parameter allows the Rock Ridge extensions to be added to areas which are not occupied by ISO 9660. As a result, the image contains an ISO 9660 file system with additional POSIX properties, which can then be processed by file systems able to understand these `rr` extensions. The added advantage of this system is that the source documents can be named arbitrarily, as both the converted ISO 9660 names together with the `rr` names now reside on the target disk.

`joliet`

Writes a disk in Joliet format (Unicode-support, long filenames).

`isolevel2`

Permits the use of file names up to 32 characters long, corresponding to ISO 9660 Interchange Level 2. Level 1 allows only the 8.3 format for normal files.

`fitcd`

With this parameter it is possible to check if the size of a resultant image would be greater than the storage capacity of a 74-minute CD (650 MB). Should this be the case, the program discontinues, thus preventing an image being generated that would not actually fit onto a CD.

If the output from `iso9660` is piped directly to `cdglow` this parameter is not necessary, as the latter carries out a similar check before writing to the CD and will not start if the image would be too big.

`maxsize=`<*bytes*> (decimal oder hexadecimal)

This parameter has the same functionality as '`fitcd`', except that it is possible to state the required limit explicitly. This is particulary important, e.g., for 63-minute CD-Rs, whose capacity is only 553 MB. When '`fitcd`' and '`maxsize=`<*bytes*>' are used together, only the value of '`maxsize`' is considered.

`followlinks`

Used to support UNIX symbolic links.

`norelocation`

Allows the use of rr directories which are deeper than 8 levels, which is the standard limit of ISO 9660.

`ignorefail`

Replaces unreadable files or directories with empty equivalents.

`checkfail`

Excludes unreadable files or directories from the image.

`source=`<*path*>

Indicates the path of the root directory.

`stdout=`<*file*> und `stderr=`<*file*>

Specifies the file name of the ISO 9660 image and the file name for error messages. If not specified, the data is written to the relevant data streams `stdout` and `stderr`.

`name=`<*volume_name*>

Sets the name of the medium, which is stored in the so-called “primary volume descriptor”. To avoid problems, no white space should appear in this name.

`publisher=`*“Text”*, `preparer=`*“Text”*, `applid=`*“Text”*

Specifies the publisher, the preparer and the application ID.

`ignore=`<*chars*>

This allows certain files to be excluded from the image. The program considers the original file names as though the string <*chars*> were removed. If the reslultant name is the same as that of a file which already exists, then the original file is ignored and does not appear in the image. If this value is set to `~`, e.g. then the file `~source.c` is excluded from the image if there already exists a file `source.c`. Should no such replica file names be found, then this parameter has no effect.

`exclude=`<*path*>

Allows a sub-directory to be excluded from the image.

`replace=@`<*path1*>`@`<*path2*>

Allows a sub-directory branch to be inserted or replaced.

To write both source directories `/y` and `/z` to a disk, create an empty directory `/x` and call

```
iso9660 source=/x replace=@/x/y@/y replace=@/x/z@/z
```

This will create a CD with the subdirectories `y` and `z` and their contents. This option provides lots of possibilities. This feature can be tested conveniently with hard disk images (see “Disk images on hard disk” on page 209).

With the following option, you can include all the necessary options in one file:

`options=`<*file*>

Each line of this file should contain only one option. By default, the files `.iso9660` und `iso9660.ini` are used to store such options.

Options and parameters which do not conform to ISO 9660

It must be emphasized that the use of the following options will lead to ISO 9660 images which will not be fully readable by all operating systems.

`longnames`

Permits the use of long file names.

`nicenames`

Allows any character to be used to construct the file name. However, it is better to use the option `allow=`<*charlist*> (see below).

`noversion`

Suppresses the version number.

`dir.ext`

Permits the use of '.xxx' extensions to directory names.

`omit.`

Suppresses the trailing period (.) of ISO 9660 file names.

`.by_`

Replaces leading periods (.) with underscores (_).

`allow=`<*charlist*>

By default, standard ISO 9660 permits only the use of capital letters, numbers, periods and underscores. Small letters must therefore be capitalized and any characters which are not allowed are converted to the underscore.'_'.

This option allows you to extend the list of characters which can appear in the resultant image. An example is `allow=-~` which allows bothof these characters to appear infile names. By using `allow=all` or `allow=ALL` it is possible to prevent any conversion from taking place. This should not be confused with the option `nicenames`; which has a more extensive effect as it ignores a number of ISO 9660 conventions, e.g., the version number. For this reason it is preferable to use `allow=all` instead of `nicenames`.

Examples

Consider a CD recorder which is connected to the first SCSI bus with the ID 6. It will therefore be addressed by als `\\.\p0b0t6` or `/dev/iXOS_SCSI0/6`. These values can be checked out with the command `inquiry` as follows:

```
inquiry \\.\p0b0t6 or inquiry /dev/iXOS_SCSI0/6
```

This provides you with the manufacturer and the product number. As soon as you have identified the Cd recorder, you can burn a Cd with the follwing:

```
iso9660 source=\ixos | cdglow -t \\.\p0b0t6
```

or

```
iso9660 source=/ixos | cdglow -t /dev/iXOS_SCSI0/6
```

This generates an ISO 9660 image from \ixos (/ixos) and all subdirectories under this directory tree and burns a Cd with this image. To check the data transfer rate, simulate the recording with the preview mode as follows::

```
iso9660 source=\ixos | cdglow -p -t \\.\p0b0t6
```

or.

```
iso9660 source=/ixos | cdglow -p -t /dev/iXOS_SCSI0/6
```

If `iso9660` does not meet the necessary data transfer rate, use a separate file or partition:

```
iso9660 source=\ixos > \temp\image
cdglow -s \temp\image -S -t \\.\p0b0t6
```

or.

```
iso9660 source=/ixos > /tmp/image
cdglow -s /tmp/image -S -t /dev/iXOS_SCSI0/6
```

where `\temp\image` (`/tmp/image`) is the target file. The effective data transfer rate used by `cdglow` can then be reduced by using the option `-f1` or `-f2`.

The attribute of the image can be modified by a variety of `iso9660` options, e.g.:

```
iso9660 rr name=USR1 source=\usr1 > \temp\image
```

or.

```
iso9660 rr name=USR1 source=/usr1 > /tmp/image
```

This example generates an ISO 9660 image from the directory tree `\usr1` (or `/usr1`) with Rock Ridge extensions.

CD can also be copied, if you have a second drive, e.g., `\\.\p0b0t3` or. `/dev/iXOS_SCSI0/3`:

```
cdglow -v -s \\.\p0b0t3 -t \\.\p0b0t6
```

or.

```
cdglow -v -s /dev/iXOS_SCSI0/3 -t /dev/iXOS_SCSI0/6
```

These commands copy one Cd to another and verify the results. It is important, however, that the source CD is faster than target CD. If this is not the case, then the options `-f1` or `-f2` should be used to lower the speed of the recording process.

Messages

`cdglow` produces message which each begin with the time in hundreds of seconds. From these messages it is possible to see, e.g., which device was found by `cdglow`, how many blocks of data should be written, how much data has already been read into the buffer, when the recording process has ended and when the CD has been finally made ready for reading. If you use the option `-v`, then an additional verification is carried out by comparing the source and the target. Should there be a difference, the number of and identity of the blocks which differ is output. The source and target should, of course, be identical, but should any errors occur during the transfer of data over the SCSI connection, it will be indicated here. If a verification is successful, the output code should is 0.

Should an actual error occur, an error message is output. In such a case two error messages are usually output. This arises from the fact that at the beginning of the recording, `cdglow` splits into two threads witha view to maintaining the required data transfer rate. This has the advantage that if one is interrupted, then the other acts as a backup process.

Production of CDs in jukeboxes

Unlike our Jukebox-Server, `cdglow` has no knowledge of jukeboxes. However, it is possible to produce CDs with the help of this server.

Note that as of version 2.1 of iXOS-JUKEMAN, you do not need to set up two different device description files or to change the only device description file in order to reserve a recorder for writing, as you had to in previous

versions. Likewise, the CD-Rs you intend to write can be included in the set of slots administered by iXOS-JUKEMAN.

First you disable the recorder drive of your jukebox for use by the file system with the command

```
cdadm detach jb.dev -d 4
```

(see also “Command line index” on page213). This assumes that the jukebox description file is `jb.dev` and that the recorder is the fourth drive of the jukebox.

After burning the disk the drive can be reattached with the command

```
cdadm attach jb.dev -d 4
```

Next insert the recordable disks into the jukebox, which can be done using the `cdadm insert` command. For more information see “Manage disks” on page 112. Note that setting the `blanks` parameter (see “Server parameters” on page 123) to a value of 2 for this purpose will speed up this process considerably (do not forget to reset the parameter to its previous value).

If you plan to insert a large number of CD-Rs, importing them one by one may be time consuming. You can open the jukebox to perform the import manually. You must, however, detach the jukebox before you can open it. After the import, and after reattaching the jukebox, the following command must be issued:

```
cdadm rescan jb.dev
```

to update the jukebox’s memory concerning which slots are filled and which are empty (see “`cdadm rescan <device>`” on page 227). The command is only mandatory for jukeboxes with non-volatile memory, like the Pioneer 5004X or the Grundig GMS 3200 which might otherwise damage themselves. For other types of jukeboxes this command will do no harm.

After successful import of the CD-Rs you have two options depending on whether your want to write only one or two CDs or plan to burn a whole series.

Burning individual CDs

Simply place an empty CD in the recorder drive. This is achieved by issuing a command similar to the following:

```
cdadm movecd jb.dev 4 9
```

This has the effect of moving a CD from slot 9 to the recorder drive (4) of the jukebox (`jb.dev`).

The CD can now be recorded like in any other CD recorder, while still allowing other CDs to be read. An example the required command is:

```
cdglow -v -s C:\temp\image.iso -S -t \\.\p1b0t5
```

or.

```
cdglow -v -s /tmp/image.iso -S -t /dev/iXOS_SCSI1/5
```

This command burns the ISO image onto the CD found attached to SCSI-ID 5 of the second SCSI controller and carries out a verification.

The CD can finally be moved out of the recorder drive back into slot 9 with the command

```
cdadm movecd jb.dev 4
```

The Cd can then be removed with the command `cdadm remove jb.dev 9` or the following command can be used to include the recorded CD as part of the jukebox file:

```
cdadm testcd jb.dev 9
```

Alternatively the CD can be inserted into the device description file by inserting the slotnumber into the CD (`disk=...`).

Burning more than one disk form a single source

Our software contains the batch script `burncds.bat` for Windows NT and the shell script `burncds.sh` for UNIX to allow several CDs to be burned from the one source. These scripts assume that the jukebox is already connected to the server, that the recorder has been appropriately reserved for the burning process and that the corresponding slots have writeable CDs. The scripts must first be tailored to the specific system configuration, however, and this is carried out with the following command:

`burncds` *<device> <destination> <source> <slotnumbers>*

or

`burncds.sh` *<device> <destination> <source> <slotnumbers>*

where *<device>* is the name of the device description file, *<destination>* the SCSI-ID of the recorder drive and *<source>* is the source data to be

burned. This value is either the SCSI-ID of another drive or the name of the file where the ISO file system image can be found. Finally, <*slotnumbers*> indicates the a list of all the slots containing the CDs to be burned.

Assuming `jukebox.dev`, is the device description file, `C:\temp\source.iso` or `/tmp/source.iso` the file containing the ISO file system image, `\\.\p0b0t6,0` or `/dev/iXOS_SCSI0/6` the SCSI-ID of the recorder drive and that the CDs in slots 1 to 5 need to be burned, then the appropriate command is:

```
burncds jukebox.dev "\\.\p0b0t6,0"
C:\temp\source.iso 1 2 3 4 5
```

or

```
burncds.sh jukebox.dev /dev/iXOS_SCSI0/6
/tmp/source.iso 1 2 3 4 5
```

Please note that the quotation marks are necessary when running the Windows NT script. This prevents the 0 being considered as another argument, rather than the actual LUN (in NT, the comma ‘,’ is used to separate arguments).

If the source data is in another SCSI CD drive, e.g., `\\.\p1b0t3,3` or `/dev/iXOS_SCSI1/4`, and the target CDs are in the slots 17, 25 and 39, then the following command is required:

```
burncds jukebox.dev "\\.\p0b0t6,0" "\\.\p1b0t3,3" 17 25
39
```

or

```
burncds.sh jukebox.dev /dev/iXOS_SCSI0/6
/dev/iXOS_SCSI1/4 17 25 39
```

If the syntax of the typed-in command is correct, then an output similar to the following should appear on the screen:

```
Burning CD in Slot 3 of jukebox.dev with source
C:\temp\source.iso  bzw.  /dev/iXOS_SCSI1/4
```

All messages from each recording are then channelled into a file of the name <*Slot#*>`.out` and, correspondingly, there will be one such output file for each slot included in the original burn command. In addition, there also exists the script `makecd.bat`, which is called by `burncds.bat` for each slot number. These files should be checked for possible errors, e.g., concerning the transport of the CDs or buffer problems. A positive indica-

tor that everything was successful is when the files correspond to each slot are the same size or if they differ only by a one or two bytes.

Configuring burncds.sh (.bat) and makecd.bat

The above scripts must be tailored to the system configuration before being used. The specific lines which may need to be changed are indicated by the trailing comment `# CONFIGURE` or by the leading/trailing comments `rem CONFIGURE BEGIN` and `rem CONFIGURE END` gesetzt.

Such lines in `burncds.sh` include:

```
PATH=$HOME/projects/jukeman/bin:$PATH # CONFIGURE
```

In the above the path `$HOME/projects/jukeman/bin` must be replaced with the path of the actual JUKEMAN.

```
cdadm movecd $devfile 1 $i # CONFIGURE
```

If the recorder drive is not the first drive in the jukebox (i.e. not the first drive which appears in the device desrciption file) you need to replace `1` with the number of the correct drive.

```
cdglow -v -s $jukesource $glowoption -t $jukedest #
CONFIGURE
```

The recording itself is started with this line. Further options `cdglow` may also be added in here, e.g., `-f1` or `-f2`, should the speed of the source not be high enough, or the option `-b` <*size*> in order to change the size of the recorder buffer. It is also very useful to use the option `-p` in order to carry out a simulation before actually executing the genuine burn jobs.

```
cdadm movecd $devfile 1 # CONFIGURE
```

Identical to `cdadm movecd`, which is described above.

The following line in `burncds.bat` should also be examined:

```
set jukeroot=D:\jukeman
```

The path `D:\jukeman` should be replace with the actual path of the JUKEMAN directory (probably `C:\jukeman`).

In `makecd.bat` the following lines may need to be modified:

```
%jukeroot%\cdadm movecd %devfile% 1 %1
```

If the recorder drive is not the fist jukebox drive (i.e. the first drive listed in the device description file), then the parameter `1` must be replaced by the actual recorder drive number.

```
%jukeroot%\cdglow -v -b 0x400000 %glowoption% -s
%jukesource% -t %jukedest%
```

This line initiates the actual recording process Further options for `cdglow` may also be added in here, e.g., `-f1` or `-f2`, should the speed of the source not be high enough, or the option `-b` <*size*> in order to change the size of the recorder buffer (`-b 0x400000` sets the buffer to 4 MB, the default value is 8 MB).

```
%jukeroot%\cdadm movecd %devfile% 1
```

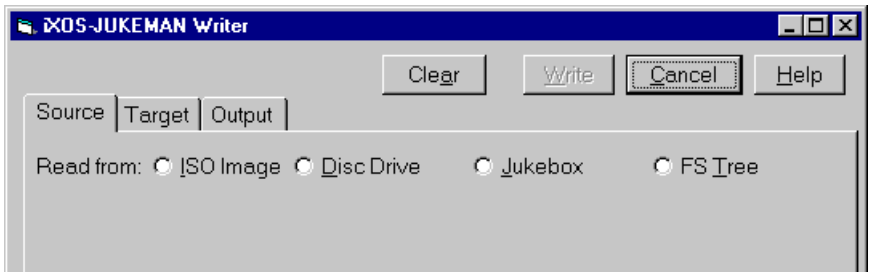
This has exactly the same effect as `cdadm movecd`.

GUI **Windows NT**

Select **[WRITE]-SINGLE TRACK AT ONCE**.

The following dialogue box appears:

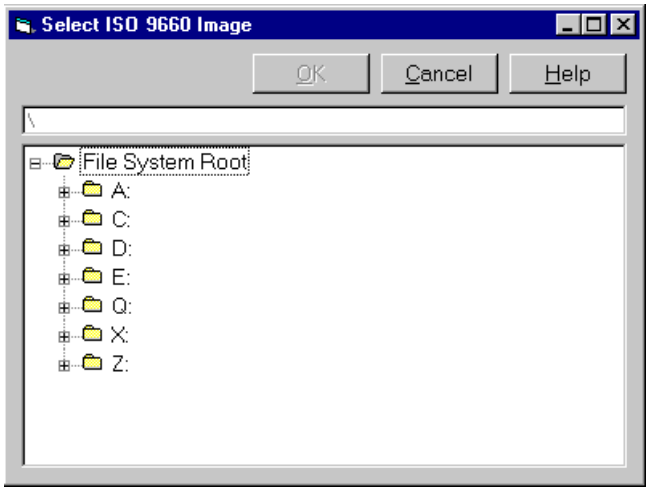
Source:



You can select one of the following sources:

ISO-Image

Choose from the selection the appropriate ISO disk image.



Disc Drive

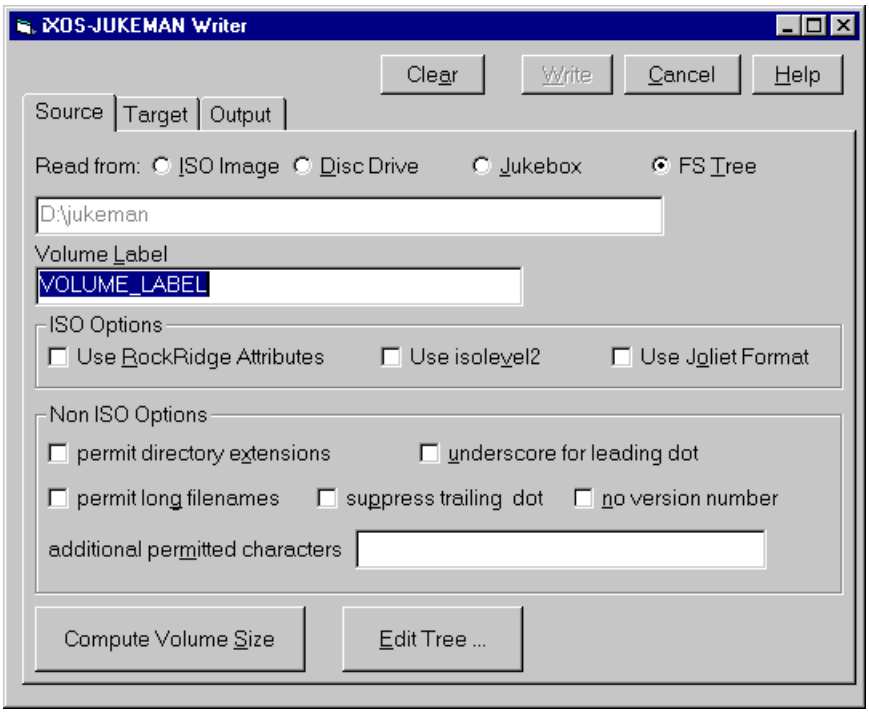
Select the appropriate source drive from the list.

Jukebox

Select the jukebox, the source drive and the required slots.

FS Tree

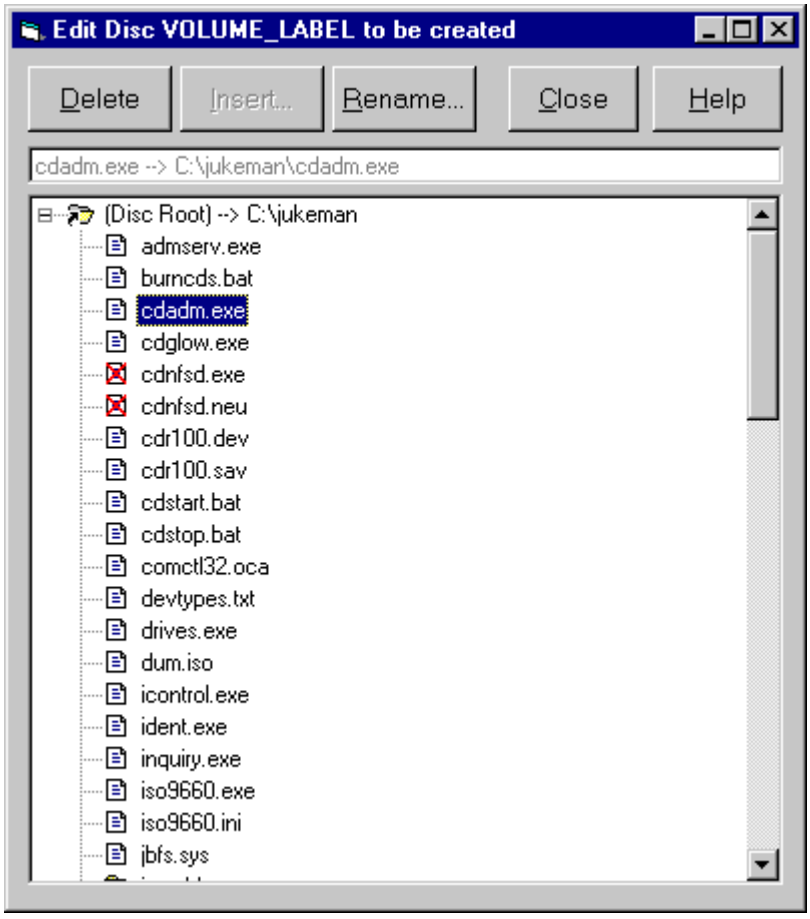
Select a directory using FS Tree. There are a variety of sources for this option:



Volume label indicates what the newly-recorded disk should be called. The **ISO Options** are described in sections “ISO 9660-conforming parameters and options” on page 136 and the **Non ISO Options** in section “Options and parameters which do not conform to ISO 9660” on page 139.

[COMPUTE VOLUME SIZE] can be used to find out the size of the disk to be recorded.

[EDIT TREE] is used to exclude, insert or rename specific files or sub-directories.:



[DELETE]: Deletes the selected file or sub-directory from the tree. The file or directory will then be marked with a red cross and be excluded from the recording.

[INSERT]: Allows a file or sub-directory to be inserted into the tree at a preselected point.

[RENAME]: Renames the selected files or sub-directories: this affects the name of the file in the target, and not the actual name in the source, which remains the same.

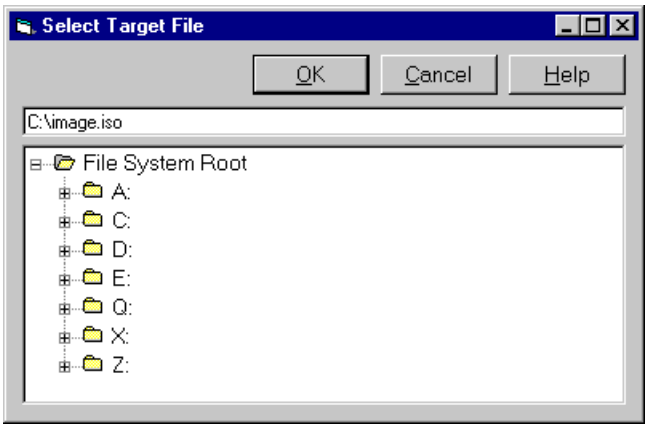
[CLOSE]: Applies the changes and closes the dialogue box.

Target:

You can choose one of the following targets:

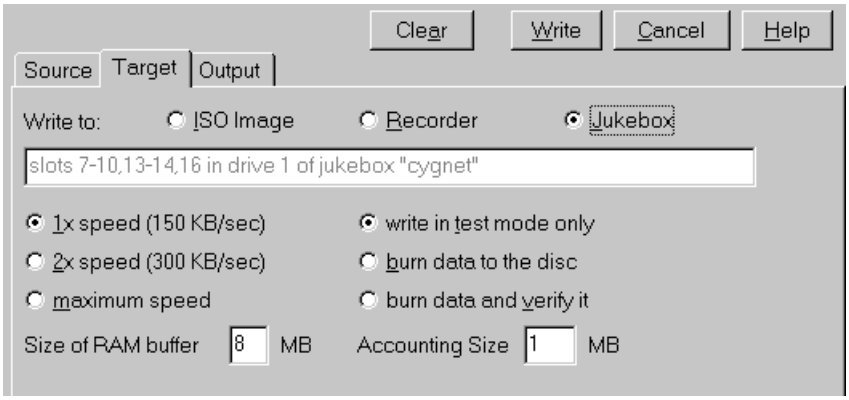
ISO-Image

Select the target directory from the list provided and type in the name for the ISO image.



Recorder

Choose the recorder drive from the list.

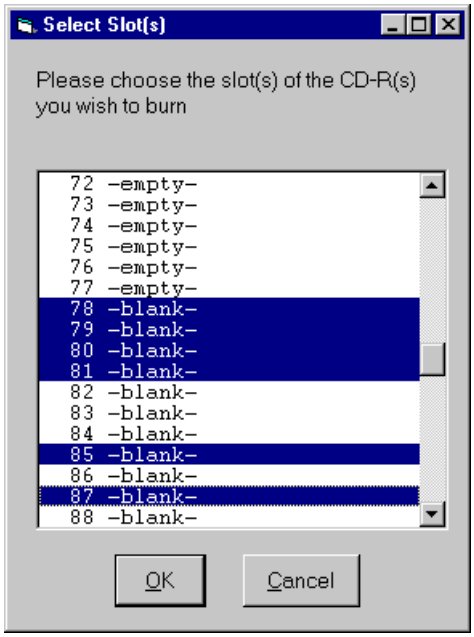


You can set the speed of the recording here (single-, dual- or maximaum speed). The option **write in test mode only** allows a simulation to be car-

ried out. Both **burn data to the disc** and **burn data and verify it** can be used to start the recording process, in the latter case also with a verification flag.

Jukebox

Choose the jukebox, recorder drive and required slots. From iXOS-JUKEMAN 2.2 and later versions, it is possible to select more than one slot, so that several CD can be burned using the same source.



The individual slots can be selected by using the [CTRL] key or, if the slots are consecutive, using [SHIFT].

When you have selected the **Source** and the **Target**, it is then possible to start the recording process with **[WRITE]**. **[CANCEL]** allows you to leave the dialogue without recording.

4.12.2 Burning disks incrementally

From version 2.0 it is possible to record disks incrementally and transparently. This means that it is possible to regard the CD as if it were a noral hard disk, once the appropriate initialisation has been carried out. The only difference is that the contents of the system buffer must at some point be explicitly burned onto the CD. It is important to note that the disk

will be written in accordance with Level 1 of the ISO 9660 file system standard. Of particular relevance is that the file names will then have conform to the 8.3 standard will original file names will be converted accordingly.

If you do not have a valid license (c.f. “Set up license keys” on page 31) for `cdglow`, which is normally stored in `writer.lic`, it is possible to write up to a maximum of 128 MB to a disk.

To allow the transparent recording of a disk, iXOS-JUKEMAN requires a global file system buffer on the hard disk. This can be entered directly into the file `server.cfg`, but can be configure in NT using the graphical interface. The relevant procedure is discribed in section “How to set up the IFS buffer” on page 46.

It is absolutely important that the data from this buffer is written to disk before the corresponding partition is modified (size, etc.) as the data will otherwise be lost. The transfer can be carried out with the `flush` command.

Tip: Please note that a disk which has been initizlised, but not yet finalized, can be read in the appropriate format in the recorder drive only by iXOS-JUKEMAN. Once the disk has been finalized, the disk can be read in any drive, even without iXOS-JUKEMAN.

The following commands pertain to the transparent recording of a disk:

```
cdadm writer action=format location=<dev>
cdadm writer [fsi=ifs] [buffer=<bufname>] action=init
location=<dev> vname=<name>
cdadm writer action=flush vname=<name>
cdadm writer action=purge vname=<name>
cdadm writer action=finalize vname=<name>
cdadm writer action=verify track=<number> vname=<name>
```

For all of the above, the speed of the recording can be set with the parameter `speed=1`, `speed=2` or `speed=4` (the default is dual speed). The size of the ring buffer can be set using the parameter `ring=<size>`, otherwise the default value of 4 MB is used.

´The following commands may be used with the command `cdadm survey -v` (or `-s`):

`+S` Size of disk including free space

`+B` Size of buffered data

`+W` Size of written data

`+w` `W+B` (data size for a disk)

`+F` `S-W` (free physical space for a disk)

`+f` `S-w` (free space for further data)

`+T` number of written tracks

All results in kB (1024 Byte)

The following is a detailed description of the commands:

`cdadm writer action=format location=`<*dev*>

Format the specified disk (not possible for CD-Rs, but for MOs, and PDs as well as hard disk images). <*dev*> is in the format <*device*>`,`<*slot*>. <*device*> is the name of the device description file and <*slot*> is the slot number.

Example:

```
cdadm writer action=format location=mo_box.dev,10
```

`cdadm writer [fsi=ifs] [buffer=`<*bufname*>`] action=init location=`<*dev*> `vname=`<*name*>

This will create an incremental file system on the specified disk. A CD-R can be written with up to 99 tracks. There is not such limitation for PDs, WOMRs, MOs and hard disk images. Please note that iXOS-JUKEMAN does not compute if there is sufficient space for the final contents track to finalize the disk. <*dev*> is in the format <*device*>`,`<*slot*>. <*device*> is the name of the device description file and <*slot*> is the slot number.

The optional parameter `fsi=ifs` specifies the incremental file system (ifs) as the the **f**ile **s**ystem **i**mplementation to be used. This is also the default setting if this parameter is not specified. Another file system implementation is the WORM file system `ixw` (see “WORM file system” on page 160).

It is possible to set up several independent buffers for the incremental file system with iXOS-JUKEMAN 2.2 (see “IFS with several independent buffers” on page 243). The parameter `buffer=`<*bufname*> must be specified if you use this set-up where <*bufname*> must be one of the buffer names defined in `server.cfg`. Each write access to the disk <*name*> will use the buffer specified during initialization.

Example:

```
cdadm writer fsi=ifs action=init location=jukebox.dev,1
vname=CDR_001
```

The incremental file system can be tested with hard disk images, even If you do not have a recorder or writeable disks (see “Disk images on hard disk” on page 209). This can be set-up using the GUI on NT or by creating a device description file (e. g., `iso.dev`), with the following contents:

Windows NT	**UNIX**
`device=image`	`device=image`
`drive=C:\temp\rfs.iso`	`drive=/tmp/rfs.iso`

The file `C:\temp\rfs.iso` or `/tmp/rfs.iso` must exist and be bigger than 64 KB (all data stored in this file will be overwritten!). When the hard disk image is attached, use the command

`cdadm writer fsi=ifs action=init location=iso.dev,1 vname=<`*name*`>`

to initialize an incremental file system for `C:\temp\rfs.iso` or `/tmp/rfs.iso`.

`cdadm writer action=flush vname=<`*name*`>`

Writes all buffered data to the disk <*name*>.

`cdadm writer action=purge vname=<`*name*`>`

Delete all buffered data for disk <*vname*>.

`cdadm writer action=finalize vname=<`*name*`>`

Finalize an incrementally written CD.

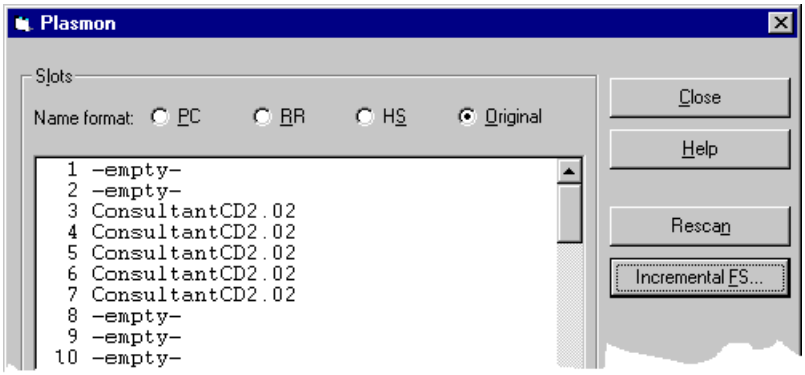
`cdadm writer action=verify track=<`*number*`> vname=<`*name*`>`

Verifies track number <*number*>. If you specify `track=all` all tracks will be verified. If no track is specified, the last written track will be verified.

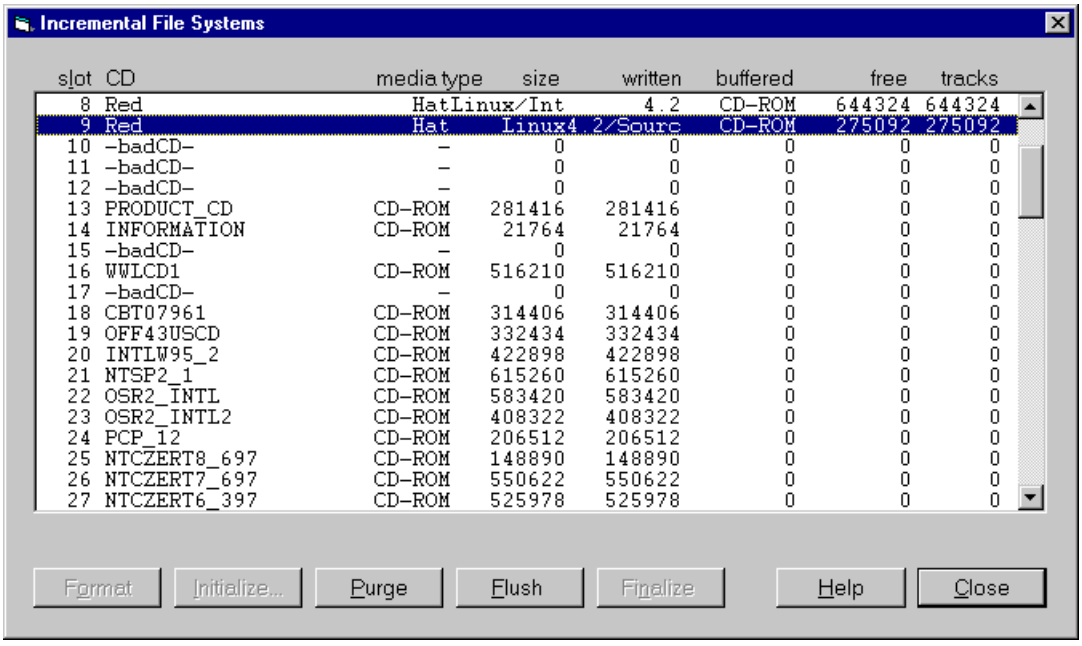
4.12.3 How to write disks incrementally

GUI **Windows NT**

Select **[CONTENTS]**:



Select **[INCREMENTAL FS...]**:



At least one separate recorder or a recorder in a jukebox enabled for IFS is needed to use the incremental file system. Alternatively, to test the functionality a hard disk image can be used (see “Disk images on hard disk” on page 209).

Format disk

Click **[FORMAT]**. Formatting of a disk may take some time.

Initialize disk

The disk needs to be initialized before files can be actually written to it. Writeable disks appear as `-badCD-` or `-blank-` in the list, deppending of the recorder. Select the disk to be initialized and click **[INITIALIZE...]**. You will be asked for the disk label, and the disk is initialized using this name.

Once the disk is initialized files can be copied to it. Everything written to the disk is stored in the IFS buffer first.

Write buffered data

Click **[FLUSH]** to actually write the buffered data to disk. This will write an additional track to the disk.

Two tracks are already used when the disk is initialized. One more track is required for the table of contents when finalizing the disk. Up to 96 tracks can be written to a CD-R, since a CD-R is limited to 99 tracks. There is no limitation to the number of tracks for PDs, WORMs, MOs, and hard disk images. Please note that iXOS-JUKEMAN does not check if sufficient space is provided on the disk to write the final contents track.

Delete buffered data

Select the disk for which the buffered data should be deleted and click **[PURGE]**.

Finalize disk

Click **[FINALIZE]** when no further data shall be written to the disk.

The disk is converted to the ISO 9660 standard level 1. It can be read in any drive, even without iXOS-JUKEMAN.

Test disk

If the disk is finished using `finalize` the disk can be made visible in the file system by testing it. See “Test disks” on page 115.

CLI **UNIX, Windows NT**

Initialize disk

```
cdadm writer fsi=ifs action=init location=jb.dev,27
vname=vol
```

This command initializes the CD-R in slot 27 of device `jb.dev` using the name "`vol`". Once the disk is initialized files can be copied to it. Everything written to the disk is stored in the IFS buffer first.

Write buffered data

To actually write the buffered data to the disk issue the command:

```
cdadm writer action=flush vname=vol
```

This will write an additional track to the disk.

Two tracks are already used when the disk is initialized. One more track is required for the table of contents when finalizing the disk. Up to 96 tracks can be written to a CD-R, since a CD-R is limited to 99 tracks. There is no limitation to the number of tracks for PDs, WORMs, MOs, and hard disk images. Please note that iXOS-JUKEMAN does not check if sufficient space is provided on the disk to write the final contents track.

Verify written data

The last written track can be verified with the command:

```
cdadm writer action=verify vname=vol
```

You can also specify the options `track=<`*number*> or `track=all` to verify track number <number> or all tracks.

Finalize disk

The disk can be finalized with the following command:

```
cdadm writer action=finalize vname=vol
```

The disk is converted to ISO 9660.

Test disk

To make the finalized disk visible in the file system, issue the following command:

```
cdadm testcd jb.dev 27
```

This instructs iXOS-JUKEMAN, to test the CD in slot 27 and add it to the file system.

4.12.4 WORM file system

Note: The WORM file system is for system integrators. End users should configure and use it with special care.

The WORM file system is a special file system implementation to write WORMs, MOs, and PDs (but not CD-Rs) incrementally and more efficiently. Compared to the IFS, files are written directly to the disks so there is not need to issue a `cdadm writer flush` command. Please note that disks written with the current version of the WORM file system can be read by iXOS-JUKEMAN only. It will be possible with a later version of iXOS-JUKEMAN however, to finalize these disks to ISO 9660.

To use the WORM file system enter a section similar to the following in the file `server.cfg`:

```
ixworm {
  maxOpenDatafiles { 10 }
  numInodes        { 250000 }
  rehashWarning    { 20 }
  HWerrorLog       { d:\temp\hw_errors.txt }
  DataFilePath     { e:\testdata }

  ixwhashdir {
    nodesize { 4 }
    files    { file1 }
    file1 {
      path { c:\temp\hashdi1 }
      size { 50 }
      mode { mapped }
    }
  }

  ixwhashname {
    nodesize { 4 }
    files    { file1 }
    file1 {
      path { e:\testdata\hashname }
      size { 50 }
      mode { mapped }
    }
  }

  ixwhashfile {
    nodesize { 4 }
    files    { file1 }
    file1 {
      path { e:\testdata\hashfile }
      size { 50 }
      mode { mapped }
    }
```

```
  }

  ixwinodes {
    nodesize { 64 }
    files    { file1 file2 }
    file1 {
      path { e:\testdata\inode1 }
      size { 200 }
      mode { file }
    }
    file2 {
      path { e:\testdata\inode2 }
      size { 200 }
      mode { file }
    }
  }
}
```

Parameter	Meaning
`maxOpenDatafiles`	Maximum number of open files for the WORM file system. This parameter selects the number of files that can be written at the same time.
`numInodes`	Total number of files/directories that can be managed by the hash tables of the WORM file systems.
`rehashWarning`	Limit for the number of rehashes, which must be exceeded to issue warnings (“file system is getting full”). Many of these warnings appear in the log files with a low (important) log level indicate the WORM file system getting full.
`HWerrorLog`	File name for the permanent log file for hardware errors.
`DataFilePath`	Path for temporary files. This directory must provide enough space to temporarily hold the data to be written to disk.
`nodesize`	Inode size in bytes. Constant 4 for Hash tables and 64 for inode- tables.
`files`	List of file labels. The hash tables and inode tables can be distributed across several files.
`path`	File name. One for list label listed under `files`.
`size`	File size in MB.
`mode`	`mapped` or `file`. Specifies if the file is created memory mapped or directly.

Using the WORM file system

If you entered the section `ixworm` in the file `server.cfg` a disk can be initialized after the next server start-up using the following command:

`cdadm writer fsi=ixw action=init location=<`*dev*`>`
`[size=<`*vsize*`>] [blksize=<`*bsize*`>] vname=<`*name*`>`

`fsi=ixw` selects the WORM file system.

<*dev*> is in the format <*device*>`,`<*slot*>. <*device*> is the name of the device description file, <*slot*> is the slot number.

The last two parameters are relevant for hard disk images only: <*vsize*> specifies the size of the image in MB. The block size <*bsize*> is 1 kB by default and must be a multiple of 512 and <= 4096.

5 Supported jukeboxes

5.1 Introduction

The hardware can be chosen from a variety of jukeboxes for CD-ROMS and writable disks. It does not matter that most of these devices are controlled in different ways.

Most functional differences are invisible for the user, but some properties have an impact on the user interface. For instance, a jukebox lacking a mail slot usually will not change disks itself upon request. For a jukebox controlled via a serial interface, the appropriate serial port has to be specified.

This chapter covers diffrent jukeboxes, their functionality, and the relevant device description files. In case you skipped the set-up chapter, please keep in mind that a lot of time can be saved by adding lines such as "`disks=1-5`" or "`save=*.sav`" to the device description files.

This chapter makes use of the common meta characters "?" (any single character) and "*" (any string) to address jukeboxes with similar names.

Note: On top of each of the following pages there is a sample device description file to minimize any problems you may have setting up the jukebox. The samples are set in two columns. Windows NT on the left, UNIX on the right.

5.2 ASM CDR????

Device type: `standard`

Windows NT	**UNIX**
`device=standard`	`device=standard`
`drive=\\.\p0b0t3`	`drive=/dev/iXOS_SCSI0/3`
`drive=\\.\p0b0t4`	`drive=/dev/iXOS_SCSI0/4`
`drive=\\.\p0b0t5`	`drive=/dev/iXOS_SCSI0/5`
`drive=\\.\p0b0t6`	`drive=/dev/iXOS_SCSI0/6`
`robot=\\.\p0b0t1`	`robot=/dev/iXOS_SCSI0/1`
`save=*.sav`	`save=*.sav`

ASM provides a range of CDR jukeboxes with capacities from 100-1563 CDs. Up to 44 CD-ROM or CD-R drives can be fitted. Only drives with caddies are supported. Please make sure to insert caddies into the mail slot so the arrow points to you, not to the jukebox. The robot is accessed through SCSI-2.

The CDRXXXX jukeboxes can be fitted with CD recorders supported by iXOS-JUKEMAN.

5.3 Cope CD Tower

Device type: `tower`

Windows NT	**UNIX**
`device=tower`	`device=tower`
`drive=\\.\p0b0t1,0`	`drive=/dev/iXOS_SCSI0/4,0`

The SCSI IDs of the Cope Tower are mapped to LUNs internally. This means you can connect up to seven towers with up to 49 drives to a single SCSI bus.

You need to enter only one drive in the device description file, the rest is done by iXOS-JUKEMAN.

You do not need to provide a save file as it has no effect.

For drives in this tower you do not have to perform a CD change by means of the GUI or `cdadm` commands anymore. All CDs can be changed manually, since all drive are checked for disk changes periodically. (see server parameter `dcheck` in “Server parameters” on page 123).

5.4 Cygnet Infinidisc/Infiniwriter

Device type: `cygnet`

Windows NT	**UNIX**
`device=cygnet`	`device=cygnet`
`drive=\\.\p0b0t4`	`drive=/dev/iXOS_SCSI0/4`
`drive=\\.\p0b0t5`	`drive=/dev/iXOS_SCSI0/5`
`robot=\\.\p0b0t1`	`robot=/dev/iXOS_SCSI0/1`
`save=*.sav`	`save=*.sav`

The Cygnet Infinidisc is a modular jukebox for CDs/CD-Rs. It supports 250 CDs (expandable to 500 CDs using a second Disk Storage Unit DSU). The number of drives can be 2 up to 8 for an Infinidisc 250 (with DSU), or 4 for an Infinidisc 500. Depending on the number of carousels the jukebox can have several mail slots.

The Infinidisc cannot be configured automatically with the Device Wizard on Windows NT as the jukebox does not provide the required information. The drives, the robot, and the device type must therefore entered “manually” in the GUI.

The Infinidisc can be fitted with recorders supported by iXOS-JUKEMAN.

5.5 Cygnet ID100

Device type: `cygnet_id100`

Windows NT	UNIX
`device=cygnet_id100`	`device=cygnet_id100`
`drive=\\.\p0b0t3`	`drive=/dev/iXOS_SCSI0/3`
`drive=\\.\p0b0t4`	`drive=/dev/iXOS_SCSI0/4`
`drive=\\.\p0b0t5`	`drive=/dev/iXOS_SCSI0/5`
`robot=\\.\p0b0t1`	`robot=/dev/iXOS_SCSI0/1`
`save=*.sav`	`save=*.sav`

The ID100 can be equipped with up to 5 packs containing 20 CDs each. Up to 4 drives can be fitted in the jukebox. The robot is addressed through SCSI-2.

The Cygnet ID100 can be fitted with recorders supported by iXOS-JUKEMAN.

5.6 Denon DRD-1408

Device type: `denon200`

Windows NT	**UNIX**
`device=denon200`	`device=denon200`
`drive=\\.\p2b0t4`	`drive=/dev/iXOS_SCSI2/4`
`drive=\\.\p2b0t5`	`drive=/dev/iXOS_SCSI2/5`
`robot=\\.\p2b0t6`	`robot=/dev/iXOS_SCSI2/6`
`save=*.sav`	`save=*.sav`

The Denon DRD-1408 jukebox for 200 CDs is equipped with two 8-speed reader drives. Disk changes are performed similar to the NSM CDR 100 by removing the packs, inserting the disks into the packs, and reinserting the packs again. The Denon jukebox does not provide a mail slot.

The DRD-1408 cannot be automatically configured with the Device Wizard on Windows NT.

5.7 DISC

Device type: `disc`

Windows NT	**UNIX**
`device=disc`	`device=disc`
`drive=\\.\p0b0t2`	`drive=/dev/iXOS_SCSI0/2`
`drive=\\.\p0b0t3`	`drive=/dev/iXOS_SCSI0/3`
`drive=\\.\p0b0t4`	`drive=/dev/iXOS_SCSI0/4`
`drive=\\.\p0b0t5`	`drive=/dev/iXOS_SCSI0/5`
`robot=\\.\p0b0t6`	`robot=/dev/iXOS_SCSI0/6`
`save=*.sav`	`save=*.sav`

DISC offers a wide range of jukeboxes with capacities that range from 250 to 1,500 CDs. The number of drives and recorders varies between 18 and 28.

The jukeboxes are controlled by one or more SCSI robots and use caddies to protect the disks. They use one or more mail slots to exchange CDs. To import a CD first put the CD in the caddy and the caddy in the mail slot. The export command puts the CD in this mail slot.

The DISC jukeboxes can be fitted with recorders supported by iXOS-JUKEMAN.

5.8 DISC DA***.*

Device type: `disc_da`

Windows NT	**UNIX**
`device=disc_da`	`device=disc_da`
`drive=\\.\p0b0t3,0`	`drive=/dev/iXOS_SCSI0/3,0`
`robot=\\.\p0b0t3,1`	`robot=/dev/iXOS_SCSI0/3,1`
`save=*.sav`	`save=*.sav`

These jukeboxes are equipped with up to 4 drives (reader or recorder drives supported by iXOS-JUKEMAN).

The DISC DA jukeboxes have a mail slot just like the NSM Mercury or the Pioneer DRM 1004X. The DISC DA is controlled through SCSI and achieves good CD change and access times.

5.9 DISC CD-CHG DJ-200/600

Device type: `disc_dj`

Windows NT	**UNIX**
`device=disc_da`	`device=disc_da`
`drive=\\.\p0b0t3,0`	`drive=/dev/iXOS_SCSI0/3,0`
`robot=\\.\p0b0t3,1`	`robot=/dev/iXOS_SCSI0/3,1`
`save=*.sav`	`save=*.sav`

The DISC CD-CHG jukeboxes contain 200 or 600 CDs and can be equipped with 2 to 6 drives (4 reader and 2 recorder drives). The robot is controlled through SCSI-2. The jukeboxes have a mail slot. The CDs are placed in packs of 50 CDs.

The DISC jukeboxes can be fitted with recorders supported by iXOS-JUKEMAN.

5.10 DSM Terastore Jukeboxen

Device type: `dsm`

Windows NT	**UNIX**
`device=dsm`	`device=dsm`
`drive=\\.\p0b0t3,0`	`drive=/dev/iXOS_SCSI0/3,0`
`robot=\\.\p0b0t3,1`	`robot=/dev/iXOS_SCSI0/3,1`
`save=*.sav`	`save=*.sav`

DSM offers jukeboxes starting from 28 up to 1,600 disks (CDs, WORMs or MOs). The number of drives is highly configurable. The disk change is performed by means of an exchange slot. Please make sure to insert the caddies into the exchange slot so the arrow points to you, not to the juke-box. The jukeboxes can be operated either by a serial RS232 line or via SCSI-2. iXOS-JUKEMAN supports the SCSI-2 variant.

The DSM jukeboxes can be fitted with recorders supported by iXOS-JUKEMAN.

5.11 ELMS DVL

Device type: `elms`

Windows NT	**UNIX**
`device=elms`	`device=elms`
`drive=\\.\p0b0t3`	`drive=/dev/iXOS_SCSI0/3`
`drive=\\.\p0b0t4`	`drive=/dev/iXOS_SCSI0/4`
`drive=\\.\p0b0t5`	`drive=/dev/iXOS_SCSI0/5`
`drive=\\.\p0b0t6`	`drive=/dev/iXOS_SCSI0/6`
`robot=\\.\p0b0t1`	`robot=/dev/iXOS_SCSI0/1`
`save=*.sav`	`save=*.sav`

The DVL can contain up to 5 packs, with 20 CDs each. Up to 4 drives can be fitted in the jukebox. The robot is controlled through SCSI-2. It is not recommended to change packs while the jukebox is running.

This jukebox has no mail slot. CDs can be changed as described in “NSM 100 CD Jukebox” on page 187.

The ELMS DVL can be fitted with recorders supported by iXOS-JUKEMAN.

5.12 Grundig GMS 1035

Device type: `grundig35`

Windows NT	**UNIX**
`device=grundig35`	`device=grundig35`
`drive=\\.\p0b0t3`	`drive=/dev/iXOS_SCSI0/3`
`drive=\\.\p0b0t4`	`drive=/dev/iXOS_SCSI0/4`
`robot=\\.\p0b0t5`	`robot=/dev/iXOS_SCSI0/5`
`save=*.sav`	`save=*.sav`

The Grundig GMS 1035 has room for 35 CDs and 2 drives, on of which can be a recorder drive.

The device type is 'grundig35'. The jukebox may have a mail slot, which is, however, hidden by a door, which you should not open during normal operation. To import/export discs you should apply the `-f` switch to the `cdadm` commands, e.g. '`cdadm import -f g35`'. With the `-f` switch, the command will return already when it is feasible to open the door (instead of waiting till the CD is tested). The `trayto` server parameter is used as a time-out for the door here; if the door is not opened *<trayto>* seconds after the `cdadm -f` call returned, normal operation is resumed.

5.13 Grundig GMS 3200

Device type: `grundig200`

Windows NT	**UNIX**
`device=grundig200`	`device=grundig200`
`drive=\\.\p1b0t3`	`drive=/dev/iXOS_SCSI1/3`
`drive=\\.\p1b0t4`	`drive=/dev/iXOS_SCSI1/4`
`drive=\\.\p1b0t5`	`drive=/dev/iXOS_SCSI1/5`
`drive=\\.\p1b0t6`	`drive=/dev/iXOS_SCSI1/6`
`robot=\\.\p1b0t2`	`robot=/dev/iXOS_SCSI1/2`
`save=*.sav`	`save=*.sav`

The Grundig jukebox GMS 3200 can take up to 200 CDs, and has four reader or recorder drives. It has a mail slot like the Mercury and a SCSI robot.

The GMS 3200 can be fitted with recorders supported by iXOS-JUKEMAN.

5.14 Grundig GMS 3280

Device type: `grundig280`

Windows NT	**UNIX**
`device=grundig280`	`device=grundig280`
`drive=\\.\p1b0t1`	`drive=/dev/iXOS_SCSI1/1`
`drive=\\.\p1b0t2`	`drive=/dev/iXOS_SCSI1/2`
`drive=\\.\p1b0t3`	`drive=/dev/iXOS_SCSI1/3`
`drive=\\.\p1b0t4`	`drive=/dev/iXOS_SCSI1/4`
`drive=\\.\p1b0t5`	`drive=/dev/iXOS_SCSI1/5`
`drive=\\.\p1b0t6`	`drive=/dev/iXOS_SCSI1/6`
`robot=\\.\p1b0t0`	`robot=/dev/iXOS_SCSI1/0`
`save=*.sav`	`save=*.sav`

The Grundig jukebox GMS 3280 contains 8 packs with 35 CDs each. Six drives can be fitted which can also be recorder drives.

The GMS 3280 can be fitted with recorders supported by iXOS-JUKEMAN.

For inserting CDs it is recommended to put the disk in the mail slot before issuing the `cdadm insert` command or to finish the import/export within 60 seconds to avoid problems with the jukebox firmware.

5.15 HP WORM/MO

Device type: `worm`

Windows NT	**UNIX**
`device=worm`	`device=worm`
`drive=\\.\p0b0t1`	`drive=/dev/iXOS_SCSI0/1`
`drive=\\.\p0b0t2`	`drive=/dev/iXOS_SCSI0/2`
`drive=\\.\p0b0t3`	`drive=/dev/iXOS_SCSI0/3`
`robot=\\.\p0b0t4`	`robot=/dev/iXOS_SCSI0/4`
`save=*.sav`	`save=*.sav`

The HP WORM/MO jukeboxes are available in all different configurations with up to 12 drives and 238 disks. The robot is controlled through SCSI-2.

5.16 Hyundai HAS-550

Device type: `hyundai`

Windows NT	**UNIX**
`device=hyundai`	`device=hyundai`
`drive=\\.\p0b0t1`	`drive=/dev/iXOS_SCSI0/1`
`robot=\\.\p0b0t4`	`robot=/dev/iXOS_SCSI0/4`
`save=*.sav`	`save=*.sav`

The Hyundai HAS-550 has the device type `hyundai` and a mail slot. This jukebox has an interesting design where the drive at the same time works as a transporter.

5.17 JVC MC-* CDROM Library

Device type: `jvc`

Windows NT	**UNIX**
`device=jvc`	`device=jvc`
`drive=\\.\p0b0t0`	`drive=/dev/iXOS_SCSI0/0`
`drive=\\.\p0b0t1`	`drive=/dev/iXOS_SCSI0/1`
`drive=\\.\p0b0t2`	`drive=/dev/iXOS_SCSI0/2`
`robot=\\.\p0b0t4`	`robot=/dev/iXOS_SCSI0/4`
`save=*.sav`	`save=*.sav`

The JVC jukeboxes contain 200 or 600 CDs and can be fitted with 2 to 6 drives (4 reader and 2 recorder drives). The robot is controlled through SCSI-2. CDs can be changed conveniently with the mail slot. The CDs are placed in packs of 50 CDs.

The JVC jukeboxes can be fitted with recorders supported by iXOS-JUKEMAN.

5.18 Kodak 100/150 CD ADL 100/150

Device type: `cdr100` bzw. `mercury`

When used with iXOS-JUKEMAN the Kodak jukeboxes are identical to the NSM CDR 100 (ADL 100) or NSM Mercury (ADL 150).

5.19 Kodak CDL 144

Device type: `kodak_cdl`

Windows NT	**UNIX**
`device=kodak_cdl`	`device=kodak_cdl`
`drive=\\.\p0b0t3`	`drive=/dev/iXOS_SCSI0/3`
`robot=\\.\p0b0t4`	`robot=/dev/iXOS_SCSI0/4`
`save=*.sav`	`save=*.sav`

The CD Library 144 jukebox can be fitted with 1 to 4 CD-ROM/CD-R drives. The drives and packs of 18 CDs allow a flexible use of the jukebox. Between 108 and 162 CDs can be loaded. The doors can be locked and a mail slot secured by a password keeps your data safe. The robot is controlled through SCSI-2. The jukebox cannot be configured using the Device Wizard on Windows NT.

Please note that the CDL 144 has been tested with firmware level 1.4b. Proper operation with later firmware revisions is not granted. It is recommended to take out the cartridges when moving the jukebox, as the disks are not locked in position and may fall out of the cartidges.

The CDL 144 can be fitted with recorders supported by iXOS-JUKEMAN.

5.20 Kubik 240 CD Jukebox

Device type: `kubik`

Windows NT	**UNIX**
`device=kubik`	`device=kubik`
`drive=\\.\p0b0t3`	`drive=/dev/iXOS_SCSI0/3`
`drive=\\.\p0b0t4`	`drive=/dev/iXOS_SCSI0/4`
`drive=\\.\p0b0t5`	`drive=/dev/iXOS_SCSI0/5`
`drive=\\.\p0b0t6`	`drive=/dev/iXOS_SCSI0/6`
`robot=com1:`	`robot=/dev/tty0`
`save=*.sav`	`save=*.sav`

In Kubik's CDR240M 240 CDs are arranged in a flat roundabout. On the back, four CD-ROM drives are ready to catch CDs that are ejected from the carousel by four thin pushers in the middle of the carousel. An additional pusher throws CDs to a separate mail slot on the front. If you access a disk, the jukebox rotates the carousel until the CD reaches the position in front of the drive in which it must be inserted. Then the pusher throws it into the drive, the drive inserts it, and you can access the data.

The robot is controlled by a serial line, so the device description file is similar to a Mercury jukebox, but since each Kubik jukebox requires an exclusive line, you do not need to specify a robot ID.

The Kubik has a separate mail slot so importing and exporting CDs is similar to NSM's Mercury. The difference is that it is not a tray; it's simply a slot that you open manually if it is unlocked and then close it manually. Then a disk can be inserted or changed.

Note that for the Kubik jukebox `cdadm insert` and `cdadm remove` use special strategies. If you do not specify a slot number or CD name, the server chooses a slot with the goal of keeping the roundabout balanced.

5.21 MDI CD 150

Device type: `mercury`

As related to iXOS-JUKEMAN, this jukebox is identical to the NSM Mercury.

5.22 Nakamichi 7 CD MCD-1020

Device type: `nakamichi` or `nec`

Windows NT	**UNIX**
`device=nakamichi`	`device=nakamichi`
`drive=\\.\p0b0t2,0`	`drive=/dev/iXOS_SCSI0/2,0`
`save=nec.sav`	`save=nec.sav`

The small Nakamichi changer is a good choice for evaluating iXOS-JUKEMAN in demo mode. Up to 7 disks can be inserted into this changer.

5.23 Nakamichi 4 CD MJ-4.8s

Device type: `nakamichi` oder `nec`

```
device=nec            device=nec
drive=\\.\p0b0t4,0    drive=/dev/iXOS_SCSI0/4,0
save=*.sav            save=*.sav
```

This 4 CD changer with an 8x drive, suitable as an internal device, is very well suited as a fast device in a hierarchical jukebox system.

5.24 Nakamichi MJ-5.16si

Device type: `nakamichi`

Windows NT	**UNIX**
`device=nakamichi`	`device=nakamichi`
`drive=\\.\p0b0t3,0`	`drive=/dev/iXOS_SCSI0/3,0`
`save=*.sav`	`save=*.sav`

The Nakamichi MJ-5.16si is a 5 CD changer with a 16x drive.

5.25 NSM 100 CD Jukebox

Device type: `cdr100`

Windows NT	**UNIX**
`device=cdr100`	`device=cdr100`
`drive=\\.\p0b0t4`	`drive=/dev/iXOS_SCSI0/4`
`robot=com1:`	`robot=/dev/ttya`
`robid=4`	`robid=4`
`save=cdr100.sav`	`save=cdr100.sav`

The CDR100 offers a capacity of 100 CDs and a high performance due to its small size.

The devices are controlled efficiently by serial lines using a special protocol which allows up to 16 devices to be daisy-chained. The robot ID `robid` is a number between 0 and 15, so that up to 16 devices can be controlled by a single serial line.

Starting with version 2.1, you can combine the settings for the robot and the robot ID using the syntax `robot=`*<serial port>*`,`*<robid>* which can be used instead of two separate lines `robot` and `robid`. For example,

`robot=com1:,4` bzw. `robot=/dev/ttya,4`

With the CDR100, use the same number for SCSI-ID and robot ID, especially if you have more than one (up to seven) devices. Refer to the CDR100 manual for information about changing the robot ID.

The CDR100 has a single drive and no mail slot, so you cannot instruct the server to export a disk to a mail slot. You change CDs by detaching the jukebox, changing the CDs manually, closing the jukebox, attaching the jukebox to the server, and calling:

`cdadm testcd` *<device>* *<list>*

which instructs the server to inspect the slots enumerated in *<list>*.

If you cannot detach the jukebox because you want to change CDs while the file system is in use and you do not want to cause errors, use `cdadm insert` and `cdadm remove`, to block user requests.

The `cdadm insert` *<device>* and `cdadm remove` *<device>* commands instruct the server to free the drive in the jukebox, so you can open the

door and change disks without impacting the clients. Clients do not receive errors, but they must wait for a response. As soon as you finish changing the CDs, close the door and tell the server which CDs you changed and to resume to normal operation. For example after you blocked the jukebox using the command

```
cdadm insert <device>
```

and changed the CDs in slots 2, 3, 4, 5, and 8, call the command

```
cdadm testcd <device> 2-5,8
```

which instructs the server to inspect slots 2-5 and 8 and to resume normal operation, including execution of all requests for the jukebox that were blocked while you changed CDs. Be sure to change the CDs quickly so users do not have to wait for responses longer than necessary.

The CDR100 can be fitted with recorders supported by iXOS-JUKEMAN.

5.26 NSM 150 CD Mercury 20/31/40

Device type: `mercury`

Windows NT	**UNIX**
`device=mercury`	`device=mercury`
`drive=\\.\p0b0t3`	`drive=/dev/iXOS_SCSI0/3`
`drive=\\.\p0b0t4`	`drive=/dev/iXOS_SCSI0/4`
`drive=\\.\p0b0t5`	`drive=/dev/iXOS_SCSI0/5`
`drive=\\.\p0b0t6`	`drive=/dev/iXOS_SCSI0/6`
`robot=com2:,0`	`robot=/dev/ttyb,0`
`save=mercury.sav`	`save=mercury.sav`

iXOS-JUKEMAN fully uses the Mercury's parallel capabilities and effectively servers 14 requests per minute for different CDs with a single Mercury. The four drives make it a good choice for archives in whicht high throughput and fast response times to many client requests are key requirements.

The Mercury is simple to use. Its CD tray enables the server to import and export CDs without mistakes, and three removeable magazines for 50 CDs each allow to quickly exchange 50 CDs, 100 CDs, or all 150 CDs.

The Mercury changer is controlled by a serial line with a capacity for up to 16 jukeboxes (even for mixed use with CDR100 jukeboxes on the same line). Note that the front key must be locked to permit software control.

The order of drives in the device description file must be the same as the order of the drive numbers (not the order of the SCSI IDs). On the back of newer Mercury 31/40 (20) jukeboxes there are four (two) switches to set the SCSI IDs. Older Mercury 31/40 models use IDs 3 to 6 corresponding to drives 1 to 4; Mercury 20 models use IDs 3 and 4 for the two drives. The first drive in the device description file must be drive 1 of the jukebox, and so on.

Note that as of iXOS-JUKEMAN version 2.1 you can combine the two lines for the serial line and the robot ID by using the syntax `robot=`<*serial port*>`,`<*robid*>.

The tray simplifies import and export of CDs. To import a CD, call:

`cdadm insert` <*device*>

and the jukebox opens the tray. Insert a CD and press any button or the tray, and the jukebox closes the tray. The server inspects the new CD and it appears in the server's file system views. If you want to remove a CD, call:

`cdadm remove` <*device*>

and the server moves a CD to the tray and opens it. The server preferably removes an invalid CD. If you want to choose a particular CD, you can specify its slot number or name in the default name format. For example,

```
cdadm remove device 25
```

causes the jukebox to remove the CD from slot 25, and

```
cdadm remove device x11r5
```

causes the jukebox to remove the CD named `x11r5`.

The Mercury can be fitted with recorders supported by iXOS-JUKEMAN.

Please note that the Mercury jukebox must be locked with the key switch to operate with iXOS-JUKEMAN.

For Mercury jukeboxes with Plextor drives, iXOS-JUKEMAN version 2.2 supports an accelerated read in of CDs, known as "fast toc".

5.27 NSM 150 CD Mercury 20s/31s/40s

Device type: `standard`

Windows NT	**UNIX**
`device=standard`	`device=standard`
`drive=\\.\p0b0t2`	`drive=/dev/iXOS_SCSI0/2`
`drive=\\.\p0b0t3`	`drive=/dev/iXOS_SCSI0/3`
`drive=\\.\p0b0t4`	`drive=/dev/iXOS_SCSI0/4`
`drive=\\.\p0b0t5`	`drive=/dev/iXOS_SCSI0/5`
`robot=\\.\p0b0t6`	`robot=/dev/iXOS_SCSI0/6`
`save=mercury.sav`	`save=mercury.sav`

These jukeboxes differ from the respective models without the trailing ‘s’ by using SCSI control instead of a serial line. Internally, however, the SCSI commands are transformed to serial commands. The advantage of these models is the pure SCSI connection; disadvantages include the need for another free SCSI ID for the robot, and more importantly, the throughput under high load can be worse than the serial models. This is due to inherent problems of the transformation of SCSI commands into serial commands.

5.28 NSM Satellite (serial)

Device type: `satellite`

Windows NT	**UNIX**
`device=satellite`	`device=satellite`
`drive=\\.\p0b0t3`	`drive=/dev/iXOS_SCSI0/3`
`drive=\\.\p0b0t4`	`drive=/dev/iXOS_SCSI0/4`
`drive=\\.\p0b0t5`	`drive=/dev/iXOS_SCSI0/5`
`drive=\\.\p0b0t6`	`drive=/dev/iXOS_SCSI0/6`
`robot=com2:,0`	`robot=ttyb,0`
`save=*.sav`	`save=*.sav`

The NSM Satellite combines the reliability of NSM jukeboxes with a modular design. The Satellite has a capacity of 60 to 135 CDs and can be fitted with up to four drives. It is not recommended to remove the packs while the Satellite is running.

The drives allow a faster CD read in which is supported by iXOS-JUKEMAN.

5.29 Panasonic LF-J50/100/200 CD Jukebox

Device type: `ps_lf_j`

Windows NT	**UNIX**
`device=ps_lf_j`	`device=ps_lf_j`
`drive=\\.\p0b0t4`	`drive=/dev/iXOS_SCSI0/4`
`drive=\\.\p0b0t3`	`drive=/dev/iXOS_SCSI0/3`
`drive=\\.\p0b0t2`	`drive=/dev/iXOS_SCSI0/2`
`robot=\\.\p0b0t6`	`robot=/dev/iXOS_SCSI0/6`
`save=ps.sav`	`save=ps.sav`

iXOS-JUKEMAN supports the Panasonic LF-J* jukeboxes with with double-head drives for CD-ROMs and 'Phase change' disks.

The drives can read CD-ROMs, and read and write Phase-Change disks.

Therefore, some peculiarities have to be kept in mind:

The “LUN mode” of the jukebox (DIP switch 5 on the back of the jukebox) should be set to 1. This avoids the drives to appear twice.

The jukebox contains magazines for 10 disks each. Every magazine is dedicated to either CD-ROMs or PDs. The CD-ROMs can be inserted and removed in the usual way using the mail slot. The sensitive PDs should not be removed from the jukebox.

The jukebox recognizes the magazines dedicated to PDs and does not allow these disks to be inserted or removed using the mail slot.

5.30 Pioneer 6 CD Changer

Device type: `pioneer6`

Windows NT	**UNIX**
`device=pioneer6`	`device=pioneer6`
`drive=\\.\p0b0t3`	`drive=/dev/iXOS_SCSI0/3`
`save=*.sav`	`save=*.sav`

The Pioneer 6-pack has several variations, depending on the type of drive used. DRM-602x, DRM-604x, and DRM-624x mainly differ in speed. The changer is always invisible, and it has no separate address, so the device description file does not specify a robot.

The changer has no mail slot, so importing and exporting CDs requires the same commands you use for NSM's CDR100. But instead of opening and closing a door, your must press the MAGAZINE EJECT button, turn one of the trays out of the magazine, and **insert a CD with data side up and label side down**, put the magazine back into the changer, and then issue the `testcd` command.

Use external termination if you experience SCSI problems with this changer.

5.31 Pioneer 18 CD Changer

Device type: `pioneer18`

Windows NT	**UNIX**
`device=pioneer18`	`device=pioneer18`
`drive=\\.\p0b0t3,0`	`drive=/dev/iXOS_SCSI0/3,0`
`robot=\\.\p0b0t3,1`	`robot=/dev/iXOS_SCSI0/3,1`
`save=*.sav`	`save=*.sav`

The DRM-1804x 18-CD changer uses three small 6-pack magazines. It addresses the actual changer separately, so it must be specified in the device description file. The changer is always LUN 1 of the drive.

CD import and export commands are the same as those for the Pioneer 6-CD changer or the NSM CDR100.

Use external termination if you experience SCSI problems with this changer.

5.32 Pioneer 100 CD Jukebox

Device type: `pioneer100`

Windows NT	**UNIX**
`device=pioneer100`	`device=pioneer100`
`drive=\\.\p0b0t4`	`drive=/dev/iXOS_SCSI0/4`
`drive=\\.\p0b0t3`	`drive=/dev/iXOS_SCSI0/3`
`drive=\\.\p0b0t2`	`drive=/dev/iXOS_SCSI0/2`
`robot=\\.\p0b0t6`	`robot=/dev/iXOS_SCSI0/0`
`save=p100.sav`	`save=p100.sav`

This jukebox contains up to 100 CDs and two to four drives. One drive can be a recorder (supported by iXOS-JUKEMAN) which occupies the space of two readers, hence only two more readers are feasible in this case.

The example configuration is a Pioneer DRM 1004X with 3 reader drives.

For a flawless disk change, keep the following in mind: If the DIP switches 3 and 4 are set to “OFF” (factory setting) the jukebox must be locked with the key switch. If DIP switch 3 is set to “ON” and DIP switch 4 is set to “OFF” the mail slot can be controlled by both SCSI commands and the operating keys, independent of the key switch setting.

5.33 Pioneer 500 CD Jukebox

Device type: `pioneer500`

Windows NT	**UNIX**
`device=pioneer500`	`device=pioneer500`
`drive=\\.\p0b0t2`	`drive=/dev/iXOS_SCSI0/2`
`drive=\\.\p0b0t3`	`drive=/dev/iXOS_SCSI0/3`
`drive=\\.\p0b0t4`	`drive=/dev/iXOS_SCSI0/4`
`drive=\\.\p0b0t5`	`drive=/dev/iXOS_SCSI0/5`
`robot=\\.\p0b0t6`	`robot=/dev/iXOS_SCSI0/6`
`save=*.sav`	`save=*.sav`

The Pioneer DRM-5004x offers 4 drives and room for 500 CDs. Both the robot and the drives are controlled by SCSI. So the drives can read data while the changer is moving a CD through the large jukebox.

The jukebox has a virtual mail slot which is in fact a position of the changer. Use

`cdadm -f insert` device

to move the changer halfway up and stop it. By specifying the `-f` switch the command returns as soon as the jukebox door may be opened. (which is, unfortunately, not so easy and requires some some power and/or skill). The time-out for the disk-change is specified by the `trayto` parameter (see “Server parameters” on page 123). Otherwise the jukebox resumes normal operation. Insert a CD in the changer and close the door again. iXOS-JUKEMAN waits for the door to be closed and moves the CD to an empty slot. Afterwards, the accumulated file system requests queue is worked off. As iXOS-JUKEMAN is monitoring the door status, there is no need to issue the command

`cdadm testcd` device

to tell the server the CD change is finished.

Similarly, using the command

`cdadm -f remove` device

a CD is exported with the changer. Open the door as soon as the command returns, remove the CD, and close the door to resume normal operation. You can also specify a slot number or CD name:

```
cdadm -f remove device x11r5
```

instructs the server to remove the CD named `x11r5`.

The DRM-5004x can be fitted with recorders supported by iXOS-JUKEMAN.

5.34 Plasmon D-Series

Device type: `plasmond`

Windows NT	**UNIX**
`device=plasmond`	`device=plasmond`
`drive=\\.\p0b0t2`	`drive=/dev/iXOS_SCSI0/2`
`drive=\\.\p0b0t3`	`drive=/dev/iXOS_SCSI0/3`
`robot=\\.\p0b0t1`	`robot=/dev/iXOS_SCSI0/1`
`save=*.sav`	`save=*.sav`

The Plasmon D-Series is available with room for 120, 240, or 480 slots for CDs or PDs, and can have 2, 4, or 6 drives fitted. PDs cannot be changed using the mail slot.

The magazines must be fitted in ascending order, so that at least magazine 1 must be fitted. It is recommended to fit all magazines.

5.35 Plasmon 150 CD-Jukebox

Device type: `standard`

Windows NT	**UNIX**
`device=standard`	`device=standard`
`drive=\\.\p0b0t2`	`drive=/dev/iXOS_SCSI0/2`
`drive=\\.\p0b0t3`	`drive=/dev/iXOS_SCSI0/3`
`drive=\\.\p0b0t4`	`drive=/dev/iXOS_SCSI0/4`
`drive=\\.\p0b0t5`	`drive=/dev/iXOS_SCSI0/5`
`robot=\\.\p0b0t6`	`robot=/dev/iXOS_SCSI0/6`
`save=*.sav`	`save=*.sav`

As related to iXOS-JUKEMAN the Plasmon CD150J is identical to the NSM Mercury 40s/31s range of jukeboxes.

Please note that the key switch must be locked for a flawless operation with iXOS-JUKEMAN.

5.36 Plextor Megaplex oder PX-J2200 200 CD Jukebox

Device type: `plextor200`

Windows NT	**UNIX**
`device=plextor200`	`device=plextor200`
`drive=\\.\p2b0t4`	`drive=/dev/iXOS_SCSI2/4`
`drive=\\.\p2b0t5`	`drive=/dev/iXOS_SCSI2/5`
`robot=\\.\p2b0t6`	`robot=/dev/iXOS_SCSI2/6`
`save=plextor.sav`	`save=plextor.sav`

The Plextor Jukebox provides space for 200 CDs and is fitted with 2 reader drives. CD changes are performed similar to the NSM CDR 100.

5.37 Smart and Friendly 7 CD CDJ 7004

Device type: `nakamichi` oder `nec`

As related to iXOS-JUKEMAN this changer is identical to the Nakamichi MCD-1020.

5.38 Smart and Friendly 4 CD CDJ 4008

Device type: `nakamichi` oder `nec`

As related to iXOS-JUKEMAN this changer is identical to the Nakamichi MJ-4.8s.

5.39 Sony CDZ-R360 CD Jukebox

Device type: `sony_cdz`

Windows NT	**UNIX**
`device=sony_cdz`	`device=sony_cdz`
`drive=\\.\p0b0t4,0`	`drive=/dev/iXOS_SCSI0/4,0`
`drive=\\.\p0b0t4,1`	`drive=/dev/iXOS_SCSI0/4,1`
`robot=\\.\p0b0t4,7`	`robot=/dev/iXOS_SCSI0/4,7`
`save=*.sav`	`save=*.sav`

The Sony CDZ-R360 maintains 360 CDs in a small jukebox with 2 drives. The changer and drives use different LUNs of the same SCSI-ID, so you can connect several jukeboxes to a single bus. The CDZ-R360 uses LUNs 0 and 1 for the drives and LUN 7 for the changer.

Similar to NSM's CDR100 or the small Pioneer changers, the CDZ-R360 has no mail slot. For the CDR100, CDs are changed manually; the CDZ-R360 supports CD exchange with the changer. After you open the door, the jukebox may move the first CD to the changer. Ignore this CD. Use the three small black buttons on the right to set the number of the slot in which you want to insert, export or change a disk. Press the larger Enter button. If there is a CD in the slot you chose, the jukebox moves it to the changer. Otherwise it frees the changer. Then you can remove the CD on the changer or put a new one on it. Press the Enter button again. The jukebox moves the new CD, if any, to the slot, and proceeds with the next slot. You can change a CD there or choose a new number with the three small black buttons. If you do not want to change more CDs, close the door. The jukebox frees the changer automatically. Tell the server in which slots CDs have changed by issuing the command

`cdadm testcd` <device>

for the slots involved.

5.40 Sony-CDL-2?00-?? CD Jukeboxen

Device type: `sony_cdl`

Windows NT	**UNIX**
`device=sony_cdl`	`device=sony_cdl`
`drive=\\.\p0b0t2`	`drive=/dev/iXOS_SCSI0/2`
`drive=\\.\p0b0t3`	`drive=/dev/iXOS_SCSI0/3`
`drive=\\.\p0b0t4`	`drive=/dev/iXOS_SCSI0/4`
`drive=\\.\p0b0t5`	`drive=/dev/iXOS_SCSI0/5`
`robot=\\.\p0b0t6`	`robot=/dev/iXOS_SCSI0/6`
`save=sony_cdl.sav`	`save=sony_cdl.sav`

The Sony-CDL-2100 jukeboxes provide room for 125, the CDL-2200 models for 225 CDs. The jukeboxes can be fitted with up to four drives (reader or recorder drives supported by iXOS-JUKEMAN).

The Sony-CDL-2* like the NSM Mercury or the Pioneer DRM 1004X have a mail slot. The SONY-CDL-2* is controlled through SCSI and achieves good CD change and access times.

5.41 Standard-SCSI-2 jukeboxes

Device type: `standard`

iXOS-JUKEMAN supports all standard SCSI-2 jukeboxes with a mail slot. The device description files are similar to the Plasmon jukebox (see page 200), but may vary in the number of drives. Most standard SCSI-2 jukeboxes can be fitted with recorders supported by iXOS-JUKEMAN (see “Supported hardware” on page 16).

5.42 Tower jukeboxes with no LUN support

Device type: `single`

Windows NT	**UNIX**
`device=single`	`device=single`
`drive=\\.\p1b0t0`	`drive=/dev/iXOS_SCSI1/0`
`drive=\\.\p1b0t1`	`drive=/dev/iXOS_SCSI1/1`
`drive=\\.\p1b0t2`	`drive=/dev/iXOS_SCSI1/2`
`drive=\\.\p1b0t3`	`drive=/dev/iXOS_SCSI1/3`
`drive=\\.\p1b0t4`	`drive=/dev/iXOS_SCSI1/4`
`drive=\\.\p1b0t5`	`drive=/dev/iXOS_SCSI1/5`
`drive=\\.\p1b0t6`	`drive=/dev/iXOS_SCSI1/6`

For tower jukeboxes where each drive is addressed by its own SCSI ID, the device type `single` is used. All drives must be listed.

CD changes can be performed with no interaction with the server (see server parameters `dcheck` in “Server parameters” on page 123).

5.43 Single drives

Device type: `single`

Windows NT	UNIX
`device=single`	`device=single`
`drive=\\.\p0b0t4`	`drive=dev/iXOS_SCSI0/4`

iXOS-JUKEMAN supports single drives. They are treated as small and simple jukeboxes.

Several single drives can be listed in one device description file.

CD changes can be performed manually without interacting with the server. The server will check these drives for disk changes periodically (see server parameter `dcheck` in “Server parameters” on page 123).

5.44 Disk images on hard disk

Device type: `image`

Windows NT	**UNIX**
`device=image`	`device=image`
`drive=D:\fakedisk.iso`	`drive=/fakedisk.iso`

Instead of real CD drives you can also use ISO 9660 formatted hard disk images of CDs (see also "iso9660" on page 136). If the file `D:\fakedisk.iso` or `/fakedisk.iso` is an ISO 9660 formatted disk image, it can be accessed with the above device description file. The file system of the image appears as a subdirectory in the file system of iXOS-JUKEMAN, as if it was a real CD in a real drive. Using this method you can set up small and fast CD servers running on hard disks only. For example with 20 CDs containing 200 MB each, you can set up a server by copying the CDs on a 4 GB hard disk and adding lines in the format format

`drive=<`*abbild*`>`

below

`device=image`

in the device description file.

A convenient way to copy CDs to hard disk is the program `cdglow`.

Example:

```
cdglow -s \\.\p0b0t4 -t D:\images\fakedisk.iso -T
```

or

```
cdglow -s /dev/iXOS_SCSI0/4 -t /images/fakedisk.iso -T
```

This will copy the CD in the drive with SCSI ID 4 on the first SCSI bus to the specified file.

On Windows NT even more CDs can be copied to hard disk if you use the transparent compression feature of NT. Mit NTFS you can compress a file

or a whole directory tree transparently. To do this, create a directory such as `\images` on an NTFS partition, and compress it with the file manager. All images created in this directory will be compressed automatically.

Another way to create ISO 9660 files is to use the formatting program `iso9660`. See “iso9660” on page 136.

Thus, hard disk images and transparent compression can be used to set up cost-effective and fast CD servers. Used in combination with jukeboxes the overall performance can be increased if the most frequently accessed disks are replicated as hard disk images.

5.45 Other jukeboxes

iXOS is continually adding new features and support for new jukebox types. Send mail to **support@europe.jukeman.com** to get an updated list of supported devices.

6 Command line index

6.1 Introduction

The following pages contain brief descriptions of the commands that can be passed to the program `cdadm`.

Note: For reasons of clarity the optional paramter "`-h` *<hostname>*" is left out on the following pages. Using this parameter, the iXOS-JUKEMAN server can be administered from any host where the administration client `cdadm` is installed. See section "Network administration" on page 96.

If you want to...	use this command:	page
attach a device to the server	`cdadm attach`	215
detach a device from the server	`cdadm detach`	218
query server parameters	`cdadm getpar`	219
set server parameters	`cdadm setpar`	228
insert disks into a device	`cdadm insert`	220
move disks to a drive	`cdadm movecd`	223
print out surveys	`cdadm survey`	229
remove disks from a device	`cdadm remove`	226
rename disks	`cdadm rename`	175
test, if the server is running	`cdadm null`	224
test certain slots for disks	`cdadm testcd`	233

If you want to...	use this command:	page
use the incremental file system	`cdadm writer`	182
log messages	`cdadm logmsg`	222

cdadm attach <*device*> [-d <*drive*>]

attach

Function:

Attach the specified device (or drive) to the server. The effect of this command is that the device and its volumes are controlled by iXOS-JUKEMAN. The device (or drive) cannot be accessed unless it is attached.

The parameter <*device*> represents a valid device description file in the JUKEMAN directory. These files have the extension `.dev`. The extension may be omitted in the command.

If the optional parameter `-d` <*drive*> is specified, a drive locked dynamically with `cdadm detach -d…` will be attached again. <*drive*> is the logical drive number.

To attach a device automatically when the server starts up, add it to the file `server.cfg` (see “Configuration file server.cfg” on page 237).

Example:

```
cdadm attach p18.dev
cdadm attach p18     (equivalent)
```

Error messages:

Error message	Meaning
`attach p18: No device description file *error* -1`	Device description file not found.
`attach p18: name in use *error* -1`	Device already attached.

cdadm byebye

byebye

Function:

Terminate the iXOS-JUKEMAN demon `cdnfsd`. The server will stop and accept no further requests. The server can be restarted by running `cdstart.bat` (under NT) or `cdnfsd` (under UNIX) (see “Starting iXOS-JUKEMAN” on page 87).

Example:

```
cdadm byebye
cdadm down      (Synonym)
```

Error messages:

Error message	Meaning
`RPC failed: cannot connect to server` `*error* -3`	Service not running

cdadm cvtree

cvtree

Function:

Rescan the configurable volume tree (views). This can be useful when views are changed in the `server.cfg` configuration file (see “Configuration file server.cfg” on page 237). The server does not need to be restarted, to reflect changes to the views configuration.

Example:

```
cdadm cvtree
```

Error messages:

-

cdadm detach <*device*> [-d <*drive*>]

detach

Function:

Detach the specified device (or drive) from the server. The device and its volumes are no longer controlled by iXOS-JUKEMAN. The device (or drive) cannot be accessed until it is attached again.

The parameter <*device*> represents a valid device description file in the JUKEMAN directory. These files have the extension `.dev`. The extension may be omitted in the command.

If the optional parameter `-d` <*drive*> is specified, the drive with the logical number <*drive*> will be locked dynamically, while the server is running. This locked drive will not be used for any further file requests to the server. However, it is possible to move disks into this drive with `cdadm movecd`… and to remove disks from this drive (e. g., to use the drive for writing disks).

Example:

```
cdadm detach p18.dev
cdadm detach p18     (equivalent)
```

Error messages:

Error message	Meaning
`No such device` `*error* 19`	Device is not attached or does not exist.

cdadm getpar <*key*>

getpar

Function:

Get the value of the server parameter <*key*>. Refer to the section “Server parameters” on page 245 for a list of all parameters. Server parameters may be set in the configuration file `server.cfg` or with the `cdadm setpar` command. Note that some parameters cannot be changed while the server is running. See “Static parameters” on page 126.

Example:

```
cdadm getpar loglev
```

Error messages:

Error message	Meaning
`No parameter "xyz" available *error* -1`	No or wrong <*key*> parameter given.

cdadm insert [-f] <*device*> [<*list*>]
cdadm import [-f] <*device*> [<*list*>]

insert
import

Function:

Insert a disk into the specified device.

The parameter <*device*> represents a valid device description file in the JUKEMAN directory. These files have the extension `.dev`. The extension may be omitted in the command.

The optional <*list*> parameter may be used to specify any number and selection of slots. The following syntax is used to specify a variety of slots:

`7`	slot 7
`3,6,40`	slots 3, 6 and 40
`3-7`	slots 3 through 7
`2,20-45`	slots 2 and slots 20 through 45
	(no specification): all slots available.

The specified slots will be scanned for empty slots and a single volume can be inserted into the first found empty slot. If there are no free slots, you will get an error message.

Note that for some devices like single drives and towers there is a dynamic drive check to detect disk changes (see server parameter `dcheck` in section “Server parameters” on page 123). You do not need to issue this command if you want to insert disks into these devices.

The optional `-f` switch can be used with the Pioneer DRM-500 and the Grundig GMS 1035. Using this switch, the command will return as soon as the door may be opened for a disk to be inserted. This is to prevent you from opening the jukebox door too early. If the switch is not specified, the command will wait, until the disk changes is finished.

Example:

```
cdadm insert merc 4-12
```

Error messages:

Error message	Meaning
`No such device` `*error* 19`	Specified device does not exist or is not attached.
`No good slot found` `*error* -1`	No empty slot in specified range.

cdadm logmsg <*level*> <*message*>

logmsg

Function:

Add <*message*> to the file `logfile.txt`. The <*level*> parameter is the log level of the message. The log file is described in "Log file logfile.txt" on page 249. If <*level*> goes beyond the parameter `loglev`, the message will not be logged.

Example:

```
cdadm logmsg 5 This goes to logfile.txt
```

Error messages:

-

cdadm movecd *<device> <drive> [<slot>]*

movecd

Function:

Move a disk from <*slot*> to <*drive*> or remove a disk from <*drive*>. This command will especially be useful, if you want to move a certain (writable) disk into a recorder drive to burn data on it.

The parameter <*device*> represents a valid device description file in the JUKEMAN directory. These files have the extension `.dev`. The extension may be omitted in the command.

The <*drive*> parameter is a number between 1 and the total number of drives in a jukebox (e.g., 4 in a Mercury jukebox).

If the <*slot*> parameter is specified, the disk from that slot will be moved into the specified drive. If <*slot*> is omitted or specified as 0, the disk currently placed in drive <*drive*> will be removed from that drive (to its previous slot).

Example:

```
cdadm movecd merc.dev 2 8
```

Error messages:

Error message	Meaning
`No such device` `*error* 19`	Specified device does not exist or is not attached.
`testing.dev no slot 55` `*error* -1`	Specified slot number is not valid.
`Not that many drives` `*error* -1`	Specified drive number is not valid.

cdadm null

null

Function:

Check if the server is active. If the server is active, there is no result. An error message will be produced if the server is not running.

Example:

```
cdadm null
```

Error messages:

Error message	Meaning
`RPC failed: cannot connect to server` `*error* -3`	Server not running.

cdadm rename [[-<nf>] <old> [<new>]]

rename

Function:

Rename a disk in the name format <*nf*>. There are three possible name formats (see also “Set up views” on page 67):

`pc`	PC format (8.3)
`rr`	Rock Ridge extensions
`hs`	High Sierra format

If no name format is supplied, the disk name will be changed for all name formats that match the old name specified.

A single disk may have a different name for each name format. The name format that is visible to the client is set in the `server.cfg` configuration file (see “Configuration file server.cfg” on page 237).

The <*old*> parameter is the current name of the disk in the given name format.

The <*new*> parameter is the new name of the disk. This parameter must not be an already existing name from the `volumes` database. If it is omitted, the <*old*> name will be removed from the database, provided that the disk is not in use.

If `cdadm rename` is called without any parameters, all unused volume names will be removed from the database. These names can then be assigned to other volumes (see also “Rename disks” on page 119).

Example:

```
cdadm rename -pc demo.cd1 office.cd1
cdadm rename    (removes unused volume names from the database)
```

Error messages:

Error message	Meaning
`Volume is active` `*error* -1`	You tried to remove a name of a disk that is in use.
`No such name` `*error* -1`	<*old*> name not known to iXOS-JUKEMAN.

```
cdadm remove [-f] <device> [<list>]
cdadm export [-f] <device> [<list>]
```

remove
export

Function:

Remove a volume from the specified device.

The parameter *<device>* represents a valid device description file in the JUKEMAN directory. These files have the extension `.dev`. The extension may be omitted in the command.

The optional *<list>* parameter may be used to specify any number and selection of slots. The following syntax is used to specify a variety of slots:

`7`	slot 7
`3,6,40`	slots 3, 6 and 40
`3-7`	slots 3 through 7
`2,20-45`	slots 2 and slots 20 through 45
	(no specification): all slots available.

(As an alternative for the slot number, a disk name may be specified.)

The slot range will be scanned for occupied slots and a single volume will be removed from the first found occupied slot. If there are no occupied slots, you will get an error message.

Note that for some devices like single drives and towers there is a dynamic drive check to detect disk changes (see server parameter `dcheck` in section “Server parameters” on page 123). You do not need to issue this command if you want to remove disks from these devices.

The optional `-f` switch can be used with the Pioneer DRM-500 and the Grundig GMS 1035. Using this switch, the command will return as soon as the door may be opened for a disk to be removed. This is to prevent you from opening the jukebox door too early. If the switch is not specified, the command will wait, until the disk change is finished.

Example:

```
cdadm remove merc 2,5,6
```

Error messages:

Error message	Meaning
`No such device` `*error* 19`	Specified device does not exist or is not attached.

cdadm rescan <device>

rescan

Function:

Reinitializes the internal memory of the jukebox, which stores information about which slot or drive contains which disk. This command is needed for some jukeboxes, if disks were changed manually. The jukebox would not know about these changes and could damage itself as it tries to move a disk in a slot which is presumed empty. The other way around the jukebox will not move disks into a drive if the slot is presumed empty according to the internal memory.

Execution of this command can take quite some time, since by default all slots are tested (e. g., about one hour for the Pioneer 5004 X), Alternatively, iXOS-JUKEMAN version 2.2 allows a partial rescan for the Pioneer 1004 X and 5004 X as well as the JVC MC-* and DISK CD CHG-* jukeboxes with the command "`cdadm testcd`", e. g., `cdadm testcd pi500.dev 1-20`. Please note, that the server parameter `blanks` (s. Seite 123) must be set to 0 to guarantee these test are actually done.

However, the command "`cdadm testcd`" applied to all slots of the jukebox should not be mistaken as a substitute for `cdadm rescan`: A `rescan` refers to the internal memory of the jukebox, whereas a `testcd` refers to the internal volumes database of the server.

The parameter *<device>* represents a valid device description file in the JUKEMAN directory. These files have the extension `.dev`. The extension may be omitted in the command.

Example:

```
cdadm rescan jb.dev
```

Error messages:

Error message	Meaning
`No such device` `*error* 19`	Specified device does not exist or is not attached.

cdadm setpar <key> <value>

setpar

Function:

Set the server parameter <*key*> to <*value*>. Refer to the section “Server parameters” on page 245 for a list of all parameters. Server parameters also may be set in the configuration file `server.cfg`. Note that some parameters cannot be changed while the server is running. See “Static parameters” on page 126.

Example:

```
cdadm setpar loglev 2
```

Error messages:

Error message	Meaning
`No parameter "xyz" available *error* -1`	Wrong <*key*> parameter given.

cdadm survey <surveytype> <columns> [<restrict>] [<sortby>]

survey

Function:

Print a survey based on devices, disks, or drives, depending on several parameters, which specify the columns, restrictions and sort criterion.

The following options may be specified:

<surveytype>	Meaning
-d	Survey based on devices
-v	Survey based on disks
-n	Survey based on the volumes database
-s	Survey based on the slots
-r	Survey based on the drives

<columns>	Print out...
General:	
+d	device names of attached devices
+n	Total number of slots of a device
+s	Slot number
+i	inode number in volumes database
Disks:	
+m	type of disk media (CD-ROM, HD image...)
+R	'r', if a recorder is necessary to read, 'a' otherwise
+a	'@', if the disk is in a drive, '-' otherwise
+u	'+', if the disk can be accessed, '-' otherwise
+U	time of last access to disk in seconds since 1970
+S	size of the disk, including free space (in kBytes)
+I	file system implementation (e. g., iso, hfs, ifs, ixw)
+v	file system specific informationen
+Y	'rw', if the disk is writable, 'r' otherwise
Names:	
+o	original disk name
+r	disk name in Rock Ridge format (rr)

<columns>	Print out...
+p	disk name in PC format (pc)
+h	disk name in High Sierra format (hs)
IFS*:	
+B	amount of buffered data for a volume
+W	amount of data (physically) written to the disk
+w	W+B (total amount of data for a disk)
+F	S-W (free space on physical disk)
+f	S-w (free space for further data)
+T	number of written tracks for a disk
Statistics:	
+D	amount of data (MBytes) read from a disk
+-D	like +D, but set all values to zero afterwards
+P	number of operations on a disk, i. e. read accesses with max. block size of 64kB
+-P	like +P, but set all values to zero afterwards
+M	number of movements of a volume into a drive
+-M	like +M, but set all values to zero afterwards
Drives:	
+n	SCSI address (path) of drive or file name for images
+v	vendor string
+p	product ID
+i	logical drive number
+t	drive type (0=hard disk, 4=CD recorder, 5=CD-ROM, 7=opt. drive, 8=jukebox)
+r	'r', if drive is a recorder, '-' otherwise
+f	'f', if drive is defective or missing, '-' otherwise
+u	'u', if drive is locked dynamically, '-' otherwise
* IFS = Incremental File System. Sizes in kBytes (1024 bytes).	

Possible combinations:

```
cdadm survey -d +dnRDPMr
cdadm survey -v +dsimRauUSIvYorphBWwFfTDPMf
cdadm survey -s +dsimRauUSIvYorphBWwFfTDPMf
cdadm survey -n +iuUIYorphDPM
cdadm survey -r +dnvpitrfu
```

Note: Any <*column*> name may be preceeded with `:`<*char*> to get quoted output for that parameter, where <*char*> is a single character. For instance `cdadm survey -d +:'d` lists all attached devices, where the device names will be quoted (`'dev1.dev'`, `'dev2.dev'`...).

<restrict>:

This optional parameter may be used to limit the survey output to entries specified in the format:

<*column*>=<*value*> (print all lines where <*columns*> is <*value*>)

<*column*>`!`<*name*> (print all lines where <*columns*> is not <*value*>)

Example:

```
cdadm survey -v +do m=HD-image
```

This command will print a disk-based survey with device names and original disk names for all disks with the type HD-image.

<sortby>:

This optional parameter may be used to sort the output by one or more columns:

`s:[-]<columns>`

The optional `-` reverses the sort order.

Example:

```
cdadm survey -v +do m=HD-image s:dB
```

´This command will print a list of all hard disk images sorted by name and size of buffered data.

Example:

`cdadm survey -d +d`	print a list of all attached devices
`cdadm survey -n +oPDM u=+ s:o`	print disk names and statistics. Output is limited to accessible disks (u=+) and sorted by original disk names (`s:o`)

Error messages:

Error message	Meaning
`usage: cdadm survey -d|-v|-s +..` `*error* -1`	Wrong <*reporttype*>.
`illegal column name` *x* `*error* -1`	column name *x* does not exist (see <*columns*> table) or is not allowed in conjunction with <*reporttype*>.

cdadm testcd <device> <list>

testcd

Function:

Test one or more slots of an attached device for their contents. Disks inserted manually will not be seen by the server, unless their corresponding slots are tested with this command. Depending on the jukebox type a rescan needs to be accomplished beforehand (see page 227).

The parameter *<device>* represents a valid device description file in the JUKEMAN directory. These files have the extension `.dev`. The extension may be omitted in the command.

The optional *<list>* parameter may be used to specify any number and selection of slots. The following syntax is used to specify a variety of slots:

`7`	slot 7
`3,6,40`	slots 3, 6 and 40
`3-7`	slots 3 through 7
`2,20-45`	slots 2 and slots 20 through 45

Example:

```
cdadm testcd merc.dev 4,5,10-20
```

Error messages:

Error message	Meaning
`No such device` `*error* 19`	Specified device does not exist or is not attached.

cdadm writer …

writer…

Function:

The "cdadm writer..." commands access the incremental file system. A detailed description can be found in the section “Burning disks incrementally” on page 151.

format

(1) `cdadm writer action=format location=`<*location*>
Format a PD/MO disk or a hard disk image. This may take some time.

init

(2) `cdadm writer [fsi=ifs] action=init location=`<*location*> `vname=`<*volume*>
Initialize a writable disk. This command creates a recordable file system on the specified disk. Two small tracks will be written for initialization. One more track will be written for initialization. Therefore, up to 96 tracks can be written to a CD-R, since it is limited to 99 Tracks. For PDs, WORMs, MOs and hard disk images there is no such limitation. The software will currently not check the space needed for the final table of contents track. Enough space must be left for successful finalization.

flush

(3) `cdadm writer action=flush vname=`<*volume*>
Actually write all buffered data to the specifies disk.

verify

(4) `cdadm writer action=verify [track=`<*number*>`]`
If `track=`... is not specified, the last written track will be verified, otherwise the track with number <*number*>. If `track=all` is specified, all tracks will be verified.

purge

(5) `cdadm writer action=purge vname=`<*volume*>
Purge all buffered data for the disk <*volume*>.

finalize

(6) `cdadm writer action=finalize vname=<volume>`
Finalize a disk with an incremental file system. All buffered data must be flushed from the buffer to the disk with a `cdadm writer flush` command before it can be finalized.

<*location*> is in the format <*device*>`,`<*slot*>, where <*device*> is the name of a valid device description file and <*slot*> is the slot number.

In addition the parameter `speed=1`, `speed=2` or `speed=3` can be specified to select the writing speed (default: double speed). The parameter `ring=`<*size*> selects the ring buffer size used for writing (default: 4 MB).

Example:

```
cdadm writer action=init location=jb.dev,4 vname=CDR_01
cdadm writer action=flush vname=CDR_01
cdadm writer action=verify vname=CDR_01
```

```
cdadm writer action=finalize vname=CDR_01
```

Error messages:

Error message	Meaning
`no vname ?` `*error* -1`	<*volume*> name is missing.
`illegal action type` `*error* -1`	Specified action not allowed.
`no vname burning` `*error* -1`	Wrong disk name or missing <*location*>.
`bad location` `*error* -1`	Invalid <*location*>. Maybe the slot is missing..
`p_wrfs(): bad init track` `*error* -1`	Disk has not been initialized.
`trec_init() fails` `*error* -1`	Disk could not be initialized.
`trec_reserve() fails` `*error* -1`	Track reservation did not succeed.
`cannot flush corrupted volume` `*error* -1`	Buffered data could not be written to disk.
`cannot finish track` `*error* -1`	Track could not be finished (disk could be corrupt)
`incomplete track` `*error* -1`	Incomplete Track.
`cannot read PVD from track 2` `*error* -1`	Primary Volume Descriptor could not be read.
`cannot start track 1` `*error* -1`	Track could not be started.

7 Configuration file server.cfg

7.1 Introduction

The main configuration file for iXOS-JUKEMAN is `server.cfg`. It contains sections for file system views, devices, server parameters, buffer and cache sizes, and comments. The file lives in the JUKEMAN directory. If this file does not exist under UNIX, you can create an example configuration file (see “UNIX” on page 25).

Each section of this file has a name. Following this name is a block of parameters in curly brackets `{ ... }`. This block contains named parameters, with their values again given in curly brackets. These values can be other parameters or defined values.

Example:

```
devices { list { jb1 jb2 }
          jb1 { startup { automatic } }
          jb2 { startup { manual }
}
```

This example defines the section `devices`. It contains 3 parameters (`list`, `jb1`, and `jb2`). The parameter `list` contains the values `jb1` and `jb2` (in this case two devices, which start-up behavior is to be defined in the following lines). The parameter `jb1` and `jb2` both contain a parameter `startup`, specifying the start-up behavior. For `jb1`, `startup` has the value `automatic`, i. e. the device `jb1` will be attached automatically when the server is started. jb2 must be attached manually.

7.2 The structure of server.cfg

This is a sample `server.cfg` file:

```
drive { W }
views {
  list { find_easy views_pc views_rr }
  roots {
    find_easy {
      views {
        list { a_m n_z some }
        roots {
          some {
            format { rr }
            deny { micros*}
            discs { *}
          }
          n_z {
            discs { [n-z]* }
          }
          a_m {
            discs { [a-m]* }
          }
        }
      }
      discs { *}
    }
    views_pc {
      discs { *}
    }
    views_rr {
      drive { Y }
      format { rr }
      label { UNIX }
    }
  }
}
```

This section defines 3 views: `views_rr`, `views_pc`, and `find_easy` (page 239)

```
devices {
       list { testing HardDisk Device1 }
       testing { startup { automatic } }
       HardDisk { startup { automatic } }
       Device1 { startup { manual } }
}
```

Devices (page 242)

```
parameters {
       dcheck { 300 }
        loglev { 1 }
}
```

Server parameters (page 245)

```
fsbuffer {
       file { fsbuffer }
       size { 40 }
       inodes { 10000 }
 }
```

IFS buffer for incr. writing (page 243)

```
dircache {
       file { dircache }
       size { 40 }
 }
```

Directory cache (page 245)

```
regcache {
       file { regcache }
       size { 40 }
 }
```

Data cache (page 245)

```
comment {
       This is just a short comment
}
```

Comments (page 246)

7.2.1 Views

This section defines a tree of views (see “Set up views” on page 67). The views are enclosed in the following structure in `server.cfg`:

```
[drive { <letter> }]
views {
  list { <view_1> <view_2> ... <view_n> }
  roots {
    <view_1> { ... }
    <view_2> { ... }
    ...
    <view_n> { ... }
  }
}
```

The `views` section may be preceeded by `drive { <letter> }`. All the views definded in `server.cfg` will be available as subdirectories of this drive letter <*letter*> (e. g., X) under Windows NT. The drive letter is ignored under UNIX.

The `list` section lists the names of any number of views (<*view_1*> to <*view_n*>), The `roots` section gives a definition for each view of the `list` section. The definition is a selection of the following parameters:

Parameter	Value
`format`	Name format (`pc`, `rr`, `hs`). If not specified, the name format will be inherited from the superordinate view. The default name format is `pc`.
`discs`	Visible disks (`*` for all). See “Table 4 - ” on page 69.
`deny`	Excluded disks. See “Table 4 - ” on page 69.
`drive`	Drive letter under Windows NT (will be ignored under UNIX). If not specified, the view will not be assigned a drive letter.
`label`	Label for the drive letter under Windows NT (Default: `JUKEMAN`, ignored under UNIX). May contain octal escape sequences like “\040” for a space.
`raw`	The `raw { 1 }` parameter selects a view format in which all disks are represented through the raw file system. You do not see the directories and files of the disks, but the full disk as a large file. The directory structure is explained in “Raw filesystem” on page 240.

If a view is not to contain disks but subviews, its definition is another `views` section. In the next example, the view `overview` contains a number of subviews:

```
views {
  list { overview }
```

```
  roots {
    overview {
      [drive { <letter> }]
      views {
        list { <subview_1> ... <subview_n> }
        roots {
          <subview_1> { ... }
          ...
          <subview_n> { ... }
        }
      }
    }
  }
}
```

Another Example:

```
drive { Z }
views {
  list { x y } roots {
    x { drive { X } discs { * } label { ALLDISCS } }
    y { drive { Y } views {
      list { a_m n_z } roots {
        a_m { discs { [a-m]* } }
        n_z { discs { [n_z]* } }
      }
    }
  }
}
```

The example above presents all disks under `x`, some under `y/a_m` and some under `/y/n_z`.

`x` and `y` are available as NT drives `X:` and `Y:`. `Z:` contains `X:` and `Y:`.

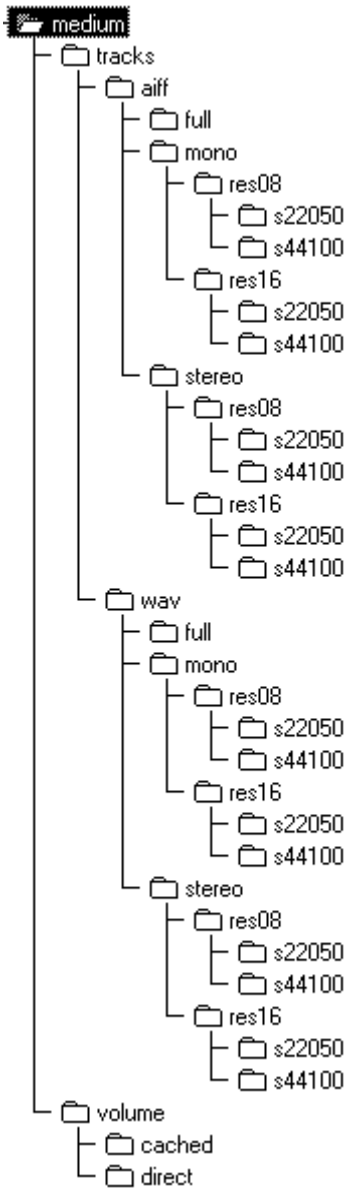
`X:` is labeled “`ALLDISCS`”, other drives have the default label “`JUKEMAN`”.

Under UNIX, presented directories are `/`, `/x`, `/y`, `/y/a_m`, `/y/n_z`.

Note, that you must create empty directories `/x`, `/y`, `/y/a_m`, `/y/n_z` and export them to see the disks after an appropriate NFS `mount` command on port 4027 (see “Intregrate iXOS-JUKEMAN into the network” on page 77).

Raw filesystem

In the raw file system, each is disk is represented through the following directory structure:



The `tracks` directory contains audio data of our experimental audio file system, which is tested for Plextor drives only and is not a component of iXOS-JUKEMAN 2.2. For more information see `medium\readme.txt`.

The `volume` directory contains several views on the disk data:

1. For read access in the `cached` directory the data cache will be used (see “The data cache” on page 40). Write access, however, will be done directly on the SCSI device or hard disk image.
2. In the `direct` directory all write and read accesses will be done directly on the SCSI device or hard disk image.

The files `stupid` and `clever` in the `cached/direct` directories both contain all data of a disk in a large file:

File	Meaning
`stupid`	Allows read and write access to the disc blocks. Used and unused block will not be distinguished.
`clever`	The raw file system will perform a more distinct error handling for `clever` files. It tries to read each block separately. Upon read failure, it asks the volume's underlying file system for unused blocks. If all unreadable blocks turn out to be unused, the raw file system succeeds and delivers zero bytes. In case of write failure on a clever file, the raw file system pretends success if all unwritable blocks are either unused or do already contain exactly the data that the user tried to write. This is useful for backups of iXOS-JUKEMAN's WORM file system, since it makes the write calls idempotent, and simplifies recovery after a crash during update of a backup through the raw file system.

The raw file system can be used, for example, to easily make ISO images on hard disk or to copy disks.

7.2.2 Devices

This section of `server.cfg` specifies, whether devices will be attached automatically at server start (setting `automatic`) or manually (setting `manual`). Devices that are not listed will not be attached. It is not required, to list all devices in this section, as they can always be attached manually with "`cdadm attach` <*device*>" (page 215). Each entry <*xyz*> presupposes a corresponding file <*xyz*>`.dev` in the JUKEMAN directory.

The GUI on Windows NT expects all devices to be listed in this section. This is to keep in mind when making changes to `server.cfg` manually.

Example:

```
devices {
      list    { p18 mercury device3 }
      p18     { startup { automatic } }
      mercury { startup { manual } }
      device3 { startup { automatic } }
}
```

This example defines 3 devices (`p18`, `mercury`, `device3`). The device names are listed in the `list { }` section. The following lines specify the

start-up behavior of the devices. Both `p18` and `device3` will be attached automatically when the server is started. `mercury` has to be attached manually.

7.2.3 Caches and buffers

Caches and buffers may be configured in the following sections. All file sizes are in MBytes. See “Set up caches and buffers” on page 36 for a detailed description.

IFS (incremental file system)

This is a buffer that temporarily holds data copied to wirtable disks with operating system commands. To actually burn the buffered data to the disk a “`cdadm writer flush`” command must be issued. The incremental file system is described in “Burning disks incrementally” on page 151.

Example:

```
fsbuffer {
      file { fsbuffer }
      size { 100 }
      inodes { 10000 }
 }
```

The example defines a buffer of 40 MB. The file name of the buffer is `fsbuffer`. The maximum number of inodes (i. e., the number of files and directories that can be stored in the buffer) is 10,000.

IFS with several independent buffers

iXOS-JUKEMAN 2.2, in addition to setting up an IFS with *one* buffer, allows setting up several independent buffers. You select one of these buffers when you initialize a disk.

The section for these buffers is `ifs { … }`. The parameter `list { }` specifies the names of all buffers (buffer names may be up to 15 characters). In the `buffers` section, the properties for each buffer are defined. The following parameters may be specified:

Parameter	Value
`file`	Contains the name of the buffer file or partition. Under UNIX a partition may be specified as a buffer (`raw{1}`). For a partition, specify the block device not the character device.

`size`	Buffer size in MBytes.
`inodes`	Maximum number of files and directories that can be stored in the buffer.
`volumes`	Maximum number of disks for this buffer.
`bsize`	Optional. Specifies the buffer block size (default: 2048 Bytes).
`raw`	Optional. If the value specified for `file` is a partition the parameter `raw { 1 }` must be added to the buffer definition.

For example, to set up two buffers replace the `fsbuffer { }`section with the following section in `server.cfg`:

```
ifs {
  list { small_buffer big_buffer }
  buffers {
    small_buffer {
      file { small.buf }
      size { 40 }
      inodes { 10000 }
      volumes { 20 }
    }
    big_buffer {
      file { big.buf }
      size { 800 }
      inodes { 300000 }
      volumes { 400 }
    }
  }
}
```

The example defines two buffers: `small_buffer` (40 MB) for up to 20 disks and 10,000 files/directories and `big_buffer` (800 MB) for up to 400 disks and 300,000 files/directories. When initializing a disk the corresponding buffer is specified in the following fashion:

```
cdadm writer fsi=ifs buffer=small_buffer action=init
location=jb.dev,5 vname=CDR_007
```

or

```
cdadm writer fsi=ifs buffer=big_buffer action=init
location=jb.dev,5 vname=CDR_007
```

Directory cache

This cache stores the directory names of the disks known to iXOS-JUKEMAN. See “The directory cache” on page 36 for a detailed description.

Example:

```
dircache {
      file { dircache }
      size { 40 }
 }
```

The example sets up a directory cache of 40 MB. The file name of the cache is `dircache`.

If the line “`file { dircache }`” is omitted, the RAM cache size will be changed.

Data cache

This cache stores data read from the disks controlled by iXOS-JUKEMAN. See “The data cache” on page 40 for a detailed description.

Example:

```
regcache {
      file { regcache }
      size { 40 }
 }
```

This example sets up a data cache of 40 MB. The file name of the cache is `regcache`.

If the line “`file { regcache }`” is omitted, the RAM cache size will be changed.

7.2.4 Server parameters

The following server parameters may be set in `server.cfg`:

(See “Server parameters” on page 123 for a full description of the parameters)

Name	Unit	Default	Min.	Max.	Function

Name	Unit	Default	Min.	Max.	Function
autodc		1	0	2	Caching behavior
blanks		0	0	2	CD type assumption/test
cdnfsp		100003	0	99999999	Program ID of NFS service
dcheck	1/100 Sec.	300	0	999999	Periodic drive checking
fullvn		0	0	1	PC name format (8.3 or 32)
hfsiso		1	0	1	Hybrid CDs: 0=HFS, 1=ISO
ignore		0	0	1	Lock server, ignore requests
iotimeo	seconds	60	0	3600	Time-out for failed disk read
jobnum		192	9	8192	Number of queue entries
loglev		4	0	9	Level of log messages
lwords		5	0	9	Internal log messages level
maxcvt		1000	10	65536	Number of nodes for views
maxthr		40	12	1024	Maximale Anzahl Threads
mdelay		3	0	99	Delay for disk changes
mountp		20000234	0	99999999	Program ID of `mount` service
nonfsd		0	0	1	Startup behav. of NFS server
portno		4027	1	65536	UDP port number
rahead		3	0	1000	Number of chunks read ahead
reject		1	0	2	Reject of incomplete disks
rtrack	Bytes	131072	8192	16777216	Data block size for caches
synclm		0	0	1	Log message buffering
trayto	seconds	60	0	99999999	Delay for mail slot close
waitpm	seconds	0	0	3600	Wait delay for portmapper

Example:

```
parameters {
      dcheck { 300 }
      loglev { 3 }
}
```

This example sets the `dcheck` server parameter to 300 and the log level `loglev` to 3 when the server starts up.

7.2.5 Comments

This section contains any comments and remarks that you wish to put in the `server.cfg` file. Everything in this section will be ignored when the

configuration file is read in by the server. Note that the comment itself must not contain a “}” .

Example:

```
comment {
      I'm just a comment.
      I will be ignored.
      You know at the end.
      That life has been short.
}
```

8 Log file logfile.txt

Normally, reading the file `logfile.txt` should not be necessary for purposes other than personal interest in internal server affairs. Tracing errors is also a reason the file should be consulted for. If you send us error reports by fax or email, please always include the relevant parts of the log file.

All messages of the server go to the log file in the JUKEMAN directory. These messages have different levels: low levels for important messages, higher levels for less important messages. The server parameter `loglev` is a limit that keeps messages with a higher level than this limit out of the log file.

- The lowest level 0 which covers all starting and stopping errors as well as hardware errors that crash the server.
- Levels 1 and 2 are for hardware error messages.
- Level 3 is for starting and stopping messages of devices.
- Level 4 is for new disks.
- Level 5 is for disk changes and other different events.
- Levels 6, 7, 8, and 9 are for error tracing only: You can use these levels, but they are far too complex to be described here in detail.

The default log level 4. Most messages up to this level and some messages from level 5 are easily comprehensible. This is a typical log message:

```
3 3/29:1137:310 @ 2 \\.\p0b0t5 is YAMAHA's CD-drive CDR400c
```

The first number is the log level. This message is about attaching a drive which is important enough for level 3.

The next number is the time in the following format:

month/date:HourMinute:SecondTenthsecond.

(months: 1-9 january-september, O=october, N=november, D=december)

The next symbol specifies the type of process that issued the command. The process symbols are:

- @ controls a drive. It actually reads the data requested by clients.
- % controls a changer. For simple jukeboxes it may also control the drive.
- ? accepts NFS requests and replies immediately or queues them.
- $ accepts internal requests to support the native file system of the NT version.
- # schedules the requests from ? and $ and passes them to % and @.
- & creates and terminates the threads @ and % upon request of #.
- - is a @ or % process during asynchronous task processing.
- ~ is a portmapper which is started if the operating system does not run a portmapper (see the description of the `waitpm` server parameter, page 123).
- ^ monitors manual disk changes in single drives and towers (see the description of the `dcheck` server parameter, page 123).
- = accepts `cdadm` requests via TCP/IP.

After the process type, a digit or letter may follow. Non-trivial requests that cannot be satisfied quickly from cache are assigned these digits or letters cyclically. This simplifies the task of tracing a single user request through the large file of a busy server. A blank indicates that no specific request caused the action. Finally, there is the individual log message. This may, for example, report that a specific piece of hardware was detected, or that the server found a disk. Disk names are reported in `rr` and `pc` format.

A large class of messages are SCSI error reports:

```
2 3/27:0329:349 @ 7 \\.\p0b0t5: 28  00  00  00  02  9b  00  00  01  00
1 3/27:0329:349 @ 7 SCSI-Error in 28 - READ (10)
1 3/27:0329:349 @ 7 status=2 sense=3 - MEDIUM ERROR
1 3/27:0329:350 @ 7 ascode 0x14 0x00 - BLOCK NOT FOUND
2 3/27:0329:355 @ 7 ReRead(0x14d800+0x800) got after 1 fault
2 3/27:0329:355 @ 7 red after 2 attempts from x11r5
```

Here the server reports an error that ocurred during a SCSI READ command on a disk. The first line reports the complete SCSI command: Ten bytes that were sent to the drive. The next line reports the command key, 28 hexadecimal, with the SCSI command name, the 10-byte version of the

SCSI READ command. The next line reports the SCSI status and sense key together with the meaning of the sense key. The next line shows the additional sense code reported by the drive and its explanation as given in the vendors SCSI manual or in the SCSI standard. If the hardware failed, this explanation should point out which part of it failed and why. The last two lines are not part of the standard SCSI error message; they report that the server retried the READ command and got the requested data in the second attempt from a disk called `x11r5`. `0x14d800+0x800` means the server tried to read 0x800 (2 KB) bytes at disk offset 0x14d800.

Relocate the log file

If you want to maintain the log file somewhere else than the JUKEMAN directory, enter the following section in the file `server.cfg`:

```
logfile {
  file { <path + file name of the log file> }
}
```

9 FAQ/Troubleshooting

9.1 Frequently asked questions (FAQ)

This is a summary of frequently asked questions about the iXOS-JUKEMAN server.

9.2 iXOS-JUKEMAN Server

I connected a new jukebox to my PC and new drive letters appeared. Why did this happen?

At startup time, NT inspects the hardware for devices which could represent file systems. For instance a normal (internal) CD drive fits perfectly well in this scheme. For SCSI devices, the type (according to the SCSI standard) is checked in this process, but unfortunately some devices don't respond properly to this request (for instance the Yamaha CDR 100 pretends to be a WORM drive). In the absence of jukeman, all devices capable of containing a file system would be displayed for instance in the file manager.

Since using a CD drive directly as well as through jukeman might yield unexpected results in some rare circumstances, and some customers would moreover like to connect a larger number of drives than can be covered by drive letters, jukeman can override this standard behavior described in the preceding paragraph in two ways.

1. If you chose to attach the corresponding device automatically on server startup (in server.cfg either by direct editing or via the GUI), then the JUKEMAN server does the necessary changes in the NT system to hide the drive letters during the next reboot. Please note that this takes effect only after a second reboot, since only then this changes get visible to the operating system!

In case that you change the startup mode again, or if the server isn't able to connect to the devices (which might be switched off for instance), the so called claim established by the automatic attach will disappear again after the second reboot.

2. If you want to get rid of the direct access by drive letters permanently, you have the choice to use the registry editor regedt32.

In

```
HKEY_LOCAL_MACHINE
        |
        --SYSTEM
            |
            --CurrentControlSet
                |
                --Services
                    genscsi
                        claims
```

you may add entries for each drive you want to get rid of.

Say for instance that (after rebooting with the connected devices) our command scsidevs yields among others an output line like

0000008 \\.\p2b0t2,0 is TOSHIBA's CD-drive “CD-ROM XM-3501TA”

and you want to hide this drive.

Then choose “Add value...” from the Edit menu and input as value the target id, where the first four characters are skipped, i.e. p2b0t2,0 in our example.

Data type should be REG_DWORD and the corresponding value, which must be supplied after OK should be 2 (with hex radix, which is the default).

You should then see an entry like

p2b0t2,0:REG_DWORD:0x2

in the right part of the window afterwards.

Please note:Using this option is a very powerful and thus dangerous action. You will most probably forget what you did in some weeks time, and this will give you (not to mention other users) a very hard time if you try to

connect another device, say a hard disk, to the claimed address, since you won't be able to use it until you remove the claim.

So you should very carefully consider what your intents are, and at least leave a note and inform your friendly local system administrator, if you choose to use this option.

Why can't my jukebox distinguish between bad CDs and empty slots?

Why are empty CD recordables labelled as bad CDs or not even recognized?

Most unfortunately some jukeboxes and/or their drives are not capable of distinguishing between empty slots and bad CDs, hence iXOS jukeman can't either there. In other words: This is a problem of your hardware, not of our software.

Likewise, the treatment of an empty CD recordable depends on the type of CD drive or writer you are using. Some drives label them as bad CD, some don't even acknowledge that there is a CD.

What does the logfile message 'CANNOT GET FH: 13 - Permission denied' mean?

This message occurs if the directory, which is created to mount on the client computers, is not exported. This means, in fact, that you didn't read the manual carefully enough. Please refer to section 'Setup file system views' in the manual for further details.

NFS mount on Solaris 2.5 or Dec UNIX

To mount the iXOS-JUKEMAN file system on Solaris 2.5, IRIX 6.* or Dec UNIX workstations an additional option is useful. The option is

`vers=2`

Example:

mount -F nfs -o port=4027,soft,retrans=14,timeo=99,vers=2 machine_name:/views/rr /mount_point

The reason is that the NFS daemon of iXOS-JUKEMAN uses NFS protocol version 2, not the newer NFS protocol version 3 on these operating systems. Please note, that omitting this option will lead to negotiations between the NFS client and our server resulting in version 2 anyway.

Why do I get weird SCSI errors, in particular data transfer overrun messages with reference to a command "[0x8 ...]" on the console of Solaris workstations?

Why do I get I/O Errors when I try to read a non root directory of a CD on Solaris?

Both errors are due to problems with the Solaris Volume Management daemon, vold. It is by default configured to control all CD drives, including those which are controlled by iXOS-JUKEMAN.

When, upon a file system request, the jukebox inserts a CD into the drive, vold detects the change and tries to automatically mount the CD - not a bad idea. Unfortunately, Sun thinks all discs containing file systems, including CDs, have 512 Byte blocks, and therefore computes too many blocks for read commands (0x8 is the hexadecimal operation code of a SCSI read command).

For example, if Solaris wants to read 64 KBytes from CD, a normal CD drive with 2048 Byte blocks must deliver 32 blocks. Sun computes that 64 KBytes are 128 Blocks and asks for them. The drive replies by sending 128 Blocks - which are 256 KBytes. Of course they do not fit into the 64 KByte buffer - therefore the error message is quite reasonable. Having detected that the CD is not mountable, vold tries to get rid of it and instructs the drive to eject the CD. The drive does it, and iXOS-JUKEMAN can no longer access the CD. Typically it was able to read the root directory (because this was faster than the failed mount), but then the CD is ejected, and iXOS-JUKEMAN cannot read anymore.

The solution depends on how brute you are. These are your options:

1. Kill vold (most brute, works immediately)

2. Delete /etc/vold.conf (prohibits that vold starts during boot)

3. Delete the line describing all CD drives from /etc/vold.conf (this is the line starting with "use cdrom...", normally placed just after the comment line "# Devices to use"

4. Edit this line so that it does not describe CDs in jukeboxes.

The last option requires that you look closely to the device names

and pathes and detect which device represents which CD drive.

Windows95 clients and slow jukeboxes

If you have lots of Windows95 clients clients competing for CDs in a slow jukebox, the LanManager Client may time out.

You can increase the client time-out by increasing the registry value
HKEY_LOCAL_MACHINE\SYSTEM\CURRENTCONTROLSET\
SERVICES\LANMANWORKSTATION\PARAMETERS\SessTimeOut
for each client.
The default is 45 (seconds). 300 is reasonable for slow jukeboxes.

How to restrict access to the JUKEMAN filesystem for certain hosts?

The restriction of the access to the JUKEMAN filesystem can be done with an option at the export of the filesystem. Such an option defines a list of hosts, which can mount the exported filesystem. But it is necessary to include the 'localhost' in the hosts list.

Example for Solaris 2.4:

The file /etc/dfs/dfstab can contain the following line:

share -F nfs -o rw=localhost:host2:host3:host4 /views/pc

License codes: unlimited and limited

Without a license code the iXOS-JUKEMAN server runs in demo mode, which means that after 2 hours the jukebox daemon has to be restarted and only the first 5 CD slots in a jukebox can be seen in the JUKEMAN filesystem.

The demo mode of the writer software allows you to burn 128MB on CD-Rs. Copying disks to hard disk images is not restricted.

To have no such restrictions you need a license code, which consists of 8 lower case letters.

There are two possibilities:

- Test license code, which expires on a certain date
- Full license code (see the software distributor list for ordering)

On UNIX platforms the license code is bound to the hostid. On Windows NT the license code is either bound to the IP address or to the network adapater card address.

Solaris: /usr/ucb/hostid

AIX: /bin/uname -m

HP-UX: /bin/uname -i

IRIX: /sbin/sysinfo -s

DEC UNIX: /sbin/ifconfig ln0

NT: Either the IP address in the result of ipconfig or the network adapter address, which can be determined by the last entry of the “Workstation active on” line of the result of 'net config workstation'

The host identifier(s) appear as wells at the top of the logfile logfile.txt

in lines with the format “Your key for a licence order:”.

For testing purposes, temporary licenses for a reasonable number of CDs

may be obtained by contacting support@usa.jukeman.com or directly from www.jukeman.com.

A server license is also bound to a certain amount of CDs one server controls. Also, one iXOS-JUKEMAN server installation can control several jukeboxes at once and so the total amount of CDs in all jukeboxes can be taken for licensing.

For the server software the license code has to be inserted in the file 'server.lic'. Example:

version=2

volumes=700

license=abcdefgh

If you got a test license code, the file 'server.lic' must also contain the expiration date. Example:

version=2

volumes=360

timeout=1996/09/30

license=abcdefgh

A license code for the writer software has to be inserted in file 'writer.lic'.

Example:

version=2

writer

license=ijklmnop

Example for test license:

version=2

writer

timeout=1996/09/30

license=ijklmnop

NFS server plus iXOS-JUKEMAN (NT)

In iXOS-JUKEMAN a NFS server is integrated. To use another NFS server for exporting filesystems on harddisk and the jukeman filesystem, some configuration has to be done, because both compete for several ressources:

- port 2049 (default for NFS services)
- port 111 for the portmapper
- anouncement as NFS service

To tell iXOS-JUKEMAN not to start the own NFS service, the file 'server.cfg' has to contain the following parameters:

```
parameters {
        portno=4027
        waitpm=300
        nonfsd=1
}
```

portno=4027 sets an alternative port for the internal communication of iXOS-JUKEMAN. nonfsd=1 causes iXOS-JUKEMAN not to register itself as NFS service.

waitpm=300 causes iXOS-JUKEMAN to try 5 minutes an inquiry for another portmapper at port 111.

PC-NFS and mount of iXOS-JUKEMAN (UNIX) filesystem

PC-NFS Pro Version 1.0 and 2.0 does not have the possibility to mount a filesystem over a different port than default (2049). If iXOS-JUKEMAN is installed on a UNIX server, per default port 4027 is used to distinguish NFS requests for the UNIX mount daemon and for iXOS-JUKEMAN.

Because it is not possible to set the port number with PC-NFS Pro, the only way to mount the jukebox filesystem is to set the port 2049 for iXOS-JUKEMAN NFS as parameter in file 'server.cfg'. But then the UNIX mount daemon cannot be used furthermore.

PC-NFS 5.1 for Windows 3.1 and WfW 3.11 has a port number option and therefore it can be used to mount the jukeman filesystem over port 4027, for instance.

command:

net use y: hostname:/views/pc /port=4027 /ro

JUKEMAN does not work properly. What shall I do?

As a general rule, you should first check this manual and the manual of your jukebox to ensure that your setup is o.k.

A good starting point for your investigations is to reboot the host jukeman runs on, while the jukebox(es) you plan to use are properly connected and powered up. At least you should restart the server.

The most useful mean for troubleshooting is the log file, logfile.txt, which you will find in the installation directory, or, more precisely, in the directory where the cdnfsd program resides. Check it out when an error occurs. Refer to section “Log messages” in the manual for details on reading the logfile.

Roughly one can distinguish three types of problems:

1. The server does not start up correctly

2. A device can not be attached

3. You can't access a CD.

Startup problems

Make sure that you don't attach any devices automatically by editing the server.cfg file either directly or via the GUI.

Check by means of the control panel that the services iXOS Admin Server and iXOS Jukebox Daemon as well as the devices iXOS generic SCSI driver and iXOS Jukebox File System have been installed and started. Another means of information might be the event log, in particular for hardware problems. If our command scsidevs does not show any SCS devices, you most probably encounter hardware problems.

You should check by an appropriate ps (ps -aux or ps -efl should do), that no cdnfsd is already running. Note that cdnfsd must either be started by root or must be suid.

Attaching problems

Attaching problems are almost always due to an incorrect device file. Check first that you chose the right device type. For each SCSI id listed as drive you should verify by our command inquiry that you can actually reach it. A typical output should be anything like

inquiry \\.\p2b0t2,0

0000002 \\.\p2b0t2,0 is TOSHIBA's CD-drive “CD-ROM XM-3501TA”

0000003 ProRevL 1875, Firmware 07/06/95

If you encounter a “Can't open” error, you should use our command scsidevs to get a list of all known SCSI devices.

inquiry /dev/iXOS_SCSI0/4,0

0000002 /dev/iXOS_SCSI0/4,0 is TOSHIBA's CD-drive "CD-ROM XM-3501TA"

0000003 ProRevL 1875, Firmware 07/06/95

If you encounter a “Bad file number” error, you should check for known devices by

inquiry /dev/iXOS_SCSI?/*

Alternatively you may try to attach a drive in a jukebox as a single device to check if this drive is alright.

The same procedure as for drives applies to SCSI changers/robots as well, except that you can't use them as single drives of course.

If your jukebox has a changer which is adressed by a serial line, you should check for logfile errors of type either

tty_open(xxx) - No such file or directory

or in case of NSM jukeboxes

nsm_recv(): got 0 bytes garbage:

which are caused by an incorrect robot line in the device file.

Please make as well sure for NSM jukeboxes that the robid is correct - check their manual for the details.

You should check the manual of your jukebox as well to ensure that the order of drives in the device file corresponds exactly to the ordering which the changer has in mind. This is a common cause of problem, and sometimes the manual are not very valuable in this respect.

If you changed CDs in the jukebox manually or nothing else helps, you should stop the server, delete the save file of the corresponding jukebox and start the server again. This will cause the jukebox to rescan all slots.

9.3 Problems with CDs

A CD is labelled as bad CD, although it can be read perfectly well on other CD drives

In most cases this is due to the fact that you exceeded the number of CDs given in your licence file. All excess CDs simply can not be addressed by jukeman.

Moreover any CD not which does not comply to the iso9660 standard or its extensions Rock Ridge or Joliet is labelled as bad CD as well.

Note, however, that some instances of bad CD tags are due to the fact that not all jukeboxes adhere the convention that the CD is inserted label up into the slots.

The testcd takes very long

This is again a problem of your hardware. Some CD drives and recorders need much time to read a CD or to detect that it is empty. iXOS jukeman assumes that it is much better for you to wait a little till you eventually can read the CD than to return that it is a bad CD (what we can't know beforehand).

9.4 iXOS-JUKEMAN Writer

Error message: BUFFER UNDERRUN

If you are not sure about the qualities of your hardware, you should first simulate the burning by using cdglow's -p option. This option simulates the writing as closely as possible, meaning that the only missing action is the real burning on the CD. If you get a buffer underrun error in this mode, you should either lower the speed of the recorder by means of the -f options, allocate a larger buffer by means of the -b option, or try to convert or save the source to one large file on a hard disk and try again with this file as source.

Please note that the version 2.0 of cdglow for AIX 4.* did not account for a slight change in the implementation of the plock subroutine, which caused problems like failing memory locks are an ever growing size of the cdglow process. These problems have been solved in version 2.1, so please upgrade to this version.

9.5 General

How can I get support for iXOS jukeman?

If you have tried everything which was listed under section “JUKEMAN does not work properly. What shall I do?” without success, you can contact us either at support@ixos.de, Tel. +49 46005-0 or FAX +49 46005-

199 for customers outside of the US or Canada, or at jukesupport@belmont.ixos.com, Tel. 415-294-5800, FAX 415-294-5836.

Please be prepared to supply the following information: Hardware platform, operating system and version, iXOS jukeman version (can for instance be obtained from the first line in the logfile), the contents of the logfile.txt and of server.cfg and all devices which cause problems.

10 Glossar

API
Application **P**rogramming **I**nterface: a set of routines, protocols, and tools for building software applications.

CLI
Command **L**ine **I**nterface, a program that will accept and execute typed in commands. See also “Conventions” on page 12.

Client
A program that runs locally on a host and exchanges data with a server program.

Gerätebeschreibungsdatei
here: A file specifying the properties of a device (e. g., a jukebox) controlled by iXOS-JUKEMAN. The file contains the device type, the drives and the slots to be used. The extension of these files is `.dev` and they are located in the JUKEMAN directory.

GUI
Graphical **U**ser **I**nterface. A program interface that takes advantage of the computer's graphics capabilities to make the program easier to use. Well-designed graphical user interfaces can free the user from learning complex command languages. See also “Conventions” on page 12.

High Sierra
File format for → ISO 9660-conforming CD-ROMs.

IFS
Incremental **F**ile **S**ystem. A procedure used by iXOS-JUKEMAN to write optical disks trackwise. Using the IFS files can be copied, moved, or deleted from disks in the usual way.

Inode
An inode is a data structure storing file properties. Each file has an inode, describing properties such as the physical position of the file on a disk. By default an inode occupies 2048 bytes.

ISO 9660
A file system standard, defining the hierarchical directory structure for CD-ROMs definiert (also known as High Sierra agreement). ISO stands for **I**nternational **S**tandard **O**rganisation.

Joliet
File system format based on → ISO 9660. The Joliet format offers special extensions such as Unicode.

JUKEMAN-directory
The directory in which iXOS-JUKEMAN is installed.

LUN
Logical **U**nit **N**umber: A subunit of a SCSI ID. A SCSI ID can have up to 8 LUNs.

MO
Magneto-**O**ptical: Rewriteable magneto-optical disk.

NFS
Network **F**ile **S**ystem: Developed by Sun to allow computers to access files over a network as if they were on local disks; now public domain, a de-facto standard. iXOS-JUKEMAN supports NFS protocol version 2.

PD
Phase change **D**ual: Optical rewritable disk.

Roboter
here: A mechanism moving the disks of a jukebox to the drives.

Rock Ridge
File system extensions based on ISO 9660 to represent UNIX file names in ISO 9660. These extension can also contains file owner and permissions.

SCSI
Small **C**omputers **S**ystem **I**nterface: SCSI is a parallel interface ANSI standard for attaching peripheral devices to computers.

SCSI-ID
Unique ID of a SCSI device.

WORM
Write **O**nce, **R**ead **M**any: Writeable optical disk that can be written once.

11 Index