



Reliant UNIX *ONLINE Documentation*

Reliant UNIX 5.45

Configuring Virtual Disks with VDisk Lite
RM200, RM300, RM400, RM600

Edition September 2000

Copyright © 2000: Fujitsu□Siemens□Computers□GmbH
Identification: U42316-J-Z915-2-76

Copyright and Trademarks

All rights reserved. □

Delivery subject to availability; right of technical modifications reserved.

All hardware and software names used are trademarks of their respective manufacturers.

1 Preface

The concept of virtual disks (or vdisks) in Reliant UNIX provides our RM systems with the capability to considerably increase the data security and data throughput of disk systems. The disk management is independent of the applications in this case.

A virtual disk is created by combining a number of physical disks to form one or more logical disks. These virtual disks operate in the same way as a traditional physical disk and are also treated as physical partitions by the system and by user processes.

Disk access in the case of virtual disks is no longer directly tied to specific disk drives, controllers or I/O controller subsystems. A virtual disk can connect a number of different disk drives.

Extensive specialist know-how is required to configure the huge potential variety of virtual disk configurations. With this in mind, we are providing our customers with an easy-to-use tool called *VDisk Lite* – Lite in the sense of uncomplicated and user-friendly. This tool makes the configuration process as straightforward as possible for the administrator. VDisk Lite offers a user interface for comfortable configuration of virtual disks.

1.1 Summary of contents

This manual describes how to configure virtual disks with the assistance of VDisk Lite.

- The **Chapter "The VDisk Lite concept"** outlines the advantages of this tool, describing all types of virtual disks currently available in VDisk Lite. In addition, embedding in the *VConfig* component of the *WebSysAdm* user interface is also explained.
- The **Chapter "Configuration with VDisk Lite"** deals with the constraints of the configuration process and describes how it can be performed properly. The most important VDisk Lite menus are also presented.
- A sample configuration finally rounds off the manual, listing the steps required to create a number of vdisks.

1.2 Target group

This manual is written for system administrators who want to use *VDisk Lite* on their computer systems. They should be familiar with the concept of virtual disks and have experience in this area.

1.3 Changes compared to the previous manual

A tree structure has been added for displaying all vdisks. Some dialog boxes and menu options also had to be modified.

1.4 Requirements

You will need a Java-enabled browser in order to access the WebSysAdmin.

1.5 Notational conventions

In this manual, the following notational conventions are used:

<i>Italics</i>	in the main body of text denote names of files, programs, commands, variables, options and screen quotations such as input fields, text fields, menus etc.
Typewriter text	denotes system output such as error messages, messages, notes and file excerpts
Bold typewriter text	refers to user input in examples
"Quotation marks"	indicate references to other chapters or manuals
ä	manual actions performed by the user



additional information, notes and tips



warnings that must be heeded

2 The VDisk Lite concept

This tool aims to simplify the relatively complicated procedure involved in configuring virtual disks. VDisk Lite was developed to facilitate this process for administrators.

VDisk Lite is a component of the new Visual Config configuration tool (abbreviated to *VConfig* below), which itself is a component of the Web-based system administration function (*WebSysAdmin*) in Reliant UNIX.

2.1 Advantages of VDisk Lite

VDisk Lite enables convenient configuration of virtual disks via a graphical user interface. Numerous vdisks can be configured in one operation. It manages free disk areas and protects against errors occurring during configurations. Because user input can be checked immediately, VDisk Lite provides protection against inappropriate or inefficient configurations.

Former operation

Configuration was previously performed by making changes manually in the */etc/dktab file*. However, because of the number of configuration options, the significant volume of data in large systems and the lack of tools, the configuration of vdisks was a complicated task prone to errors, which was best left to specialists. Now, with the availability of *VConfig*, the configuration of virtual disks should be a much easier task.

What are the advantages of VDisk Lite?

The main advantages of VDisk Lite at a glance include:

- Excellent ease of use

The configuration of virtual disks is now easier than ever before thanks to VDisk Lite. With the availability of the graphical user interface, the administrator can now take all necessary information from the screen. Only the attributes of relevance to the administrator are shown. The interface offers a modern look and feel which makes familiarization easier.

- Protection against bad configurations

One of the best features offered by VDisk Lite is improved protection against incorrect or inefficient configurations since all input is checked immediately and rejected if errors are detected.

- High degree of transparency

The information is displayed very clearly, providing administrators with a good overview even in the case of large configurations. Effective selection options allow administrators to also refer to subareas of the system or administration domain as they make their selection. Groups of vdisks are also supported.

In its role as volume manager, VDisk Lite also manages free disk areas and prevents multiple disk assignments.

- Fast processing

The ability to configure several hundred vdisks at the same time offers considerable advantages time-wise in comparison with the traditional method where a separate configuration description was prepared for each disk.

- Domain configurations

The concept takes account from the outset of the configuration of entire administration domains. The commands and system interfaces to be used are adapted to domain configurations. All actions required to operate the vdisks are performed. Vdisk configurations are automatically distributed to all nodes in administration domains.

2.1.1 Compatibility with existing configurations

Existing vdisk configurations can be displayed by means of the new interface and can continue to run without restrictions. However, they cannot be changed with VDisk Lite.



Features such as memory disks and mirroring of root disks are not supported by VDisk Lite. These vdisk types can continue to be configured with the previous vdisk procedure, i. e. using the `/etc/dktab` file. For further details on this topic, refer to the chapter "Mirroring root disks" in the manual "Virtual Disks".

2.1.2 Available vdisk types

A number of different types of virtual disks are available:

- **Base** vdisks (initial vdisk type) cover the data area of a physical disk. They can be extended online to form *mirror* vdisks.
- **Mirror** vdisks comprise two or three physical disks that have been mirrored. This type of configuration is also referred to as RAID1. The essential purpose is to increase data security.
- **Stripe** vdisks comprise at least two vdisks of type *base* or *mirror*. This type of configuration corresponds to RAID Level 0. *Stripe* vdisks are usually used to improve performance.
- **Concat** vdisks are created by logically concatenating several *base*, *mirror* or *stripe* disks. All pieces must be the same type.
- **P-Vdisks** (partitioned virtual disks) are created when a large vdisk is split into several *p-vdisk* type disks of a required size using a partition algorithm.
- **Statesave** devices are required for mirror disks and are created automatically on import.

2.1.3 Available vdisk groups

Every vdisk is assigned to a group in VDisk Lite. There are no global group vdisks. Configuration changes can only be performed within a group. The group name must be a numeric name in the range 1 - 9999.

A distinction is made between the following types of vdisk groups:

1. Local

A vdisk group used exclusively by a node in an administration domain.

2. Observe

A vdisk group whose physical disks can be switched between several nodes (generally two). At any one time, only one host can access the vdisks in this group.

3. RMS

A vdisk group whose physical disks can be accessed by several hosts at the same time. The vdisks in this group are termed *multihosted*. The software (RMS) ensures that each vdisk is only configured on one node at any one time.

4. OPS

A vdisk group whose physical disks can be accessed by several nodes at the same time. However, the disks on the participating nodes are configured simultaneously and are used by all nodes at the same time as devices.

5. OPSRES

As OPS, but the vdisks in this group support Oracle resilvering.



Virtual disks in the Observe, OPS and RMS groups are also termed global.

When a group is created, the associated statesave device is also generated automatically.

2.2 Embedding in VConfig

VDisk Lite is a component of *VConfig*, which itself is a component of the Web-based system administration solution *WebSysAdmin*, which resides in the Reliant UNIX operating system. The *WebSysAdmin* user interface is largely self-explanatory. We will therefore provide you simply with sufficient details below for working with the user interface.



You will find more detailed information on this topic in the manual "[System Administration within a Domain \(Reliant UNIX\)](#)".

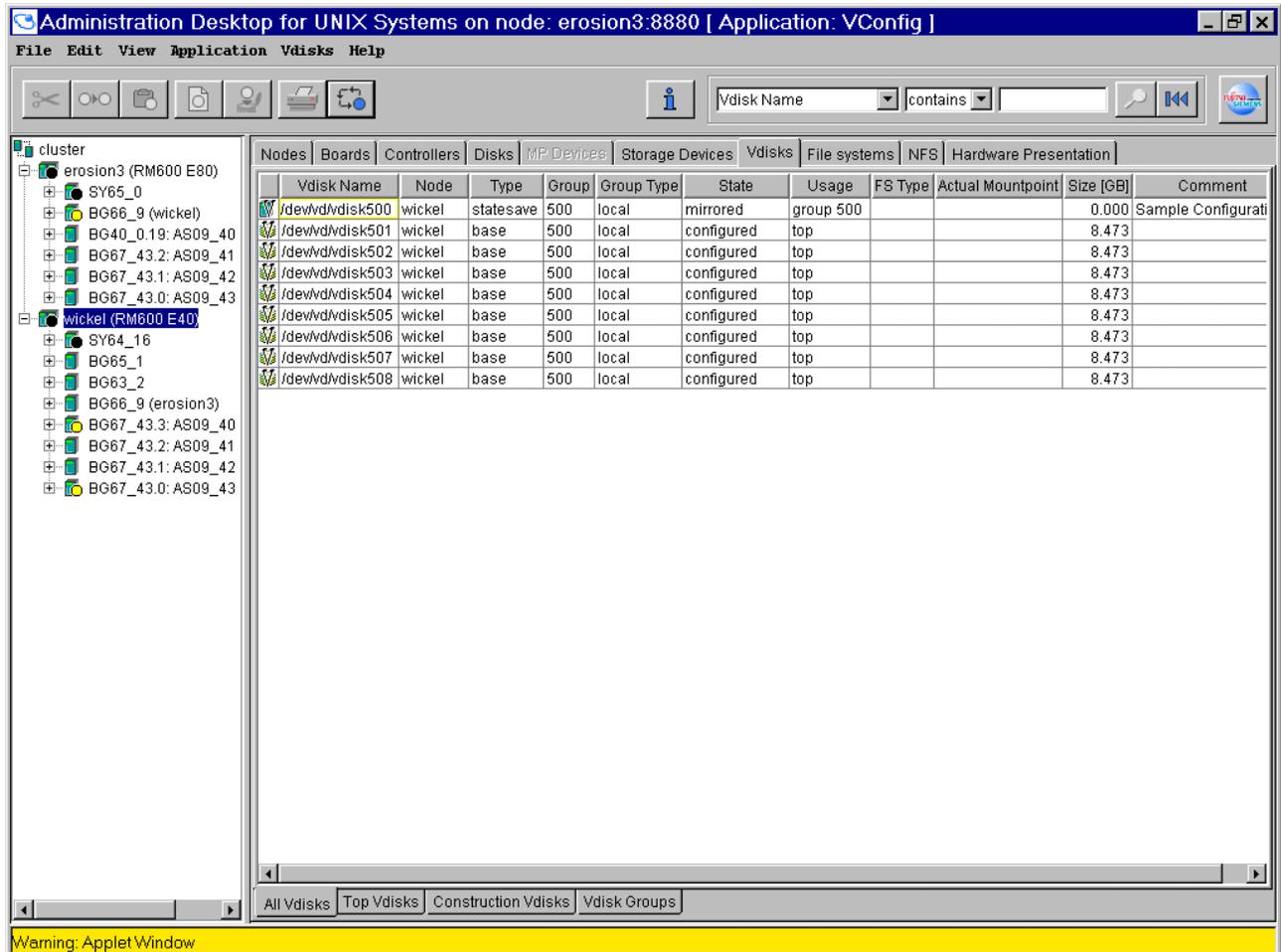


Figure 1: The VConfig user interface component (here: "Vdisks" tab)

2.2.1 User interface elements

The user interface essentially comprises the following elements:

- Menubar

The menubar at the top window frame comprises fixed and object-specific menu entries. These in turn comprise fixed and object-specific submenus.

- Fixed menu entries are always available and can always be selected (such as the *File* menu).
- Object-specific menu entries can only be selected if an object was selected beforehand and this object permits the selected action. The menu entries are grayed out if the action cannot be selected (thus for example a vdisk, for which a file system is entered in the *vfstab* file cannot be deleted).

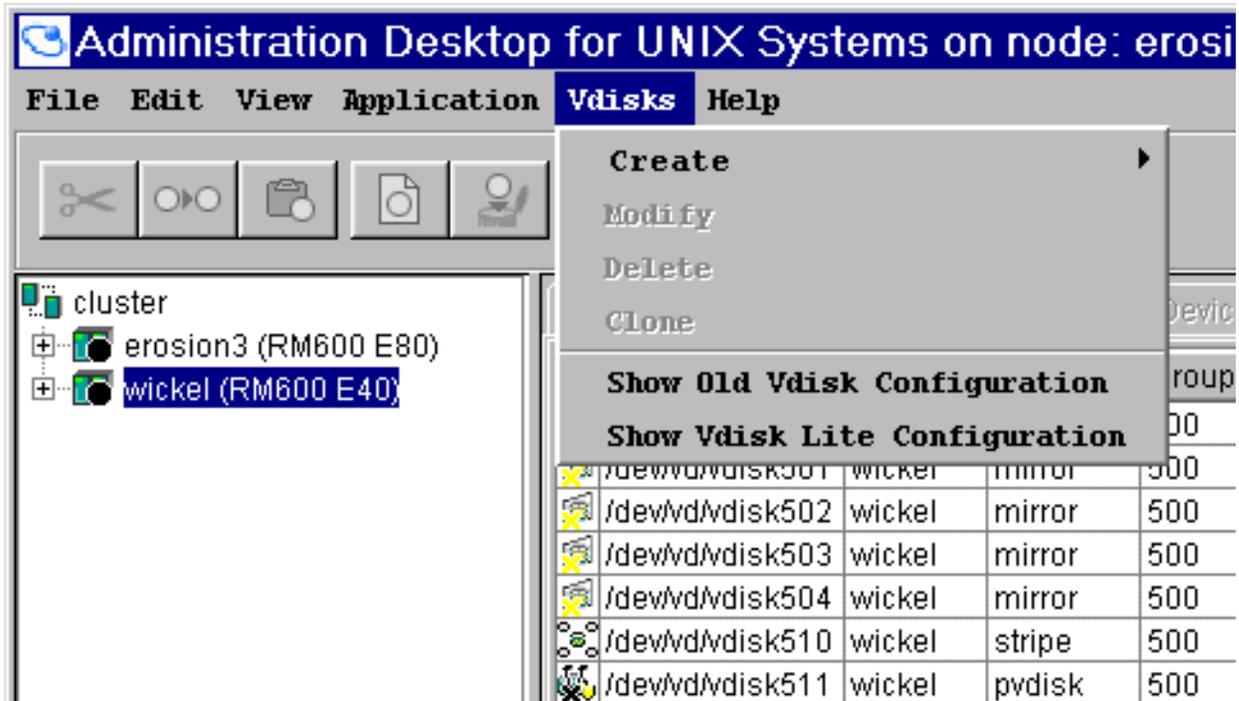


Figure 2: The VConfig menubar (approximates figure)

- Toolbar

The toolbar contains all standard buttons for some of the most frequently used functions to allow faster access to these functions. The function is displayed by holding the mouse cursor over a button for a short while without clicking on it.



Figure 3: Toolbar

Selection Configurator

You can use the selection configurator to specify selection criteria for each attribute of the displayed elements. Only objects that correspond to these selection criteria are then displayed. These criteria are used as the basic setting for the selected objects.

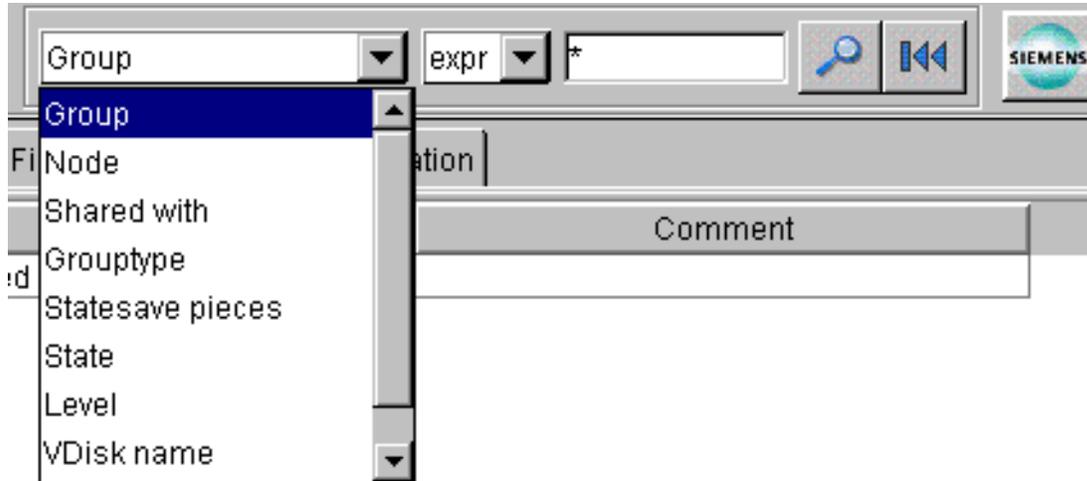


Figure 4: Selection Configuration for "Vdisks" tab

Hierarchy Browser

The hardware configuration for a node or an administration domain can be displayed with the Hierarchy Browser. The number of elements displayed in the table view grows in accordance with the degree of expansion.

In the case of vdisks, you can opt to view all vdisks on a node or in the entire administration domain. Since vdisk elements are logical elements that cannot be assigned directly to a cage, SCSI port or slot, it does not make sense here to split the elements below node level.

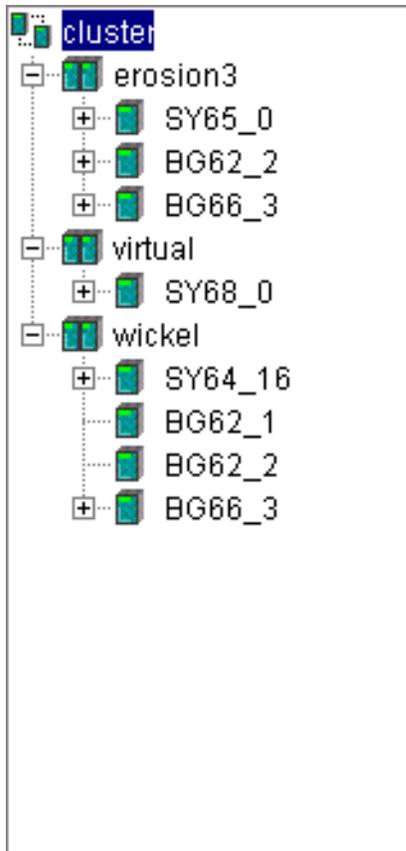


Figure 5: Hierarchy browser (approximates figure)

• Tab bar

The tab bar contains the *Vdisks* tab. Pressing this button brings you to the page for editing vdisks.



Figure 6: Tab bar

• Table view

The table view shows you all vdisks created with VDisk Lite together with their attributes.

	VDisk name	Node	Shared with	Type	Group	GroupType	Size[GB]	FsType	ActualMountPoint	State
	/dev/vd/vdisk101	erosion3	erosion3	base	100	local	4.235			not-configur
	/dev/vd/vdisk103	erosion3	erosion3	base	100	local	4.235			not-configur
	/dev/vd/vdisk104	erosion3	erosion3	base	100	local	4.235			not-configur
	/dev/vd/vdisk105	erosion3	erosion3	base	100	local	4.235			not-configur
	/dev/vd/vdisk106	erosion3	erosion3	base	100	local	4.235			not-configur
	/dev/vd/vdisk107	erosion3	erosion3	base	100	local	4.235			not-configur
	/dev/vd/vdisk108	erosion3	erosion3	base	100	local	8.473			not-configur
	/dev/vd/vdisk109	erosion3	erosion3	base	100	local	8.473			not-configur
	/dev/vd/vdisk110	erosion3	erosion3	base	100	local	4.235			not-configur

Figure 7: Table view of the "Vdisks" tab (approximates figure)

• Tree view

The tree view shows the structures in which the individual vdisks are incorporated. The dependencies on construction vdisks (e.g. nestings) are more transparent as a result.



No object-specific actions can be performed in the tree structure.

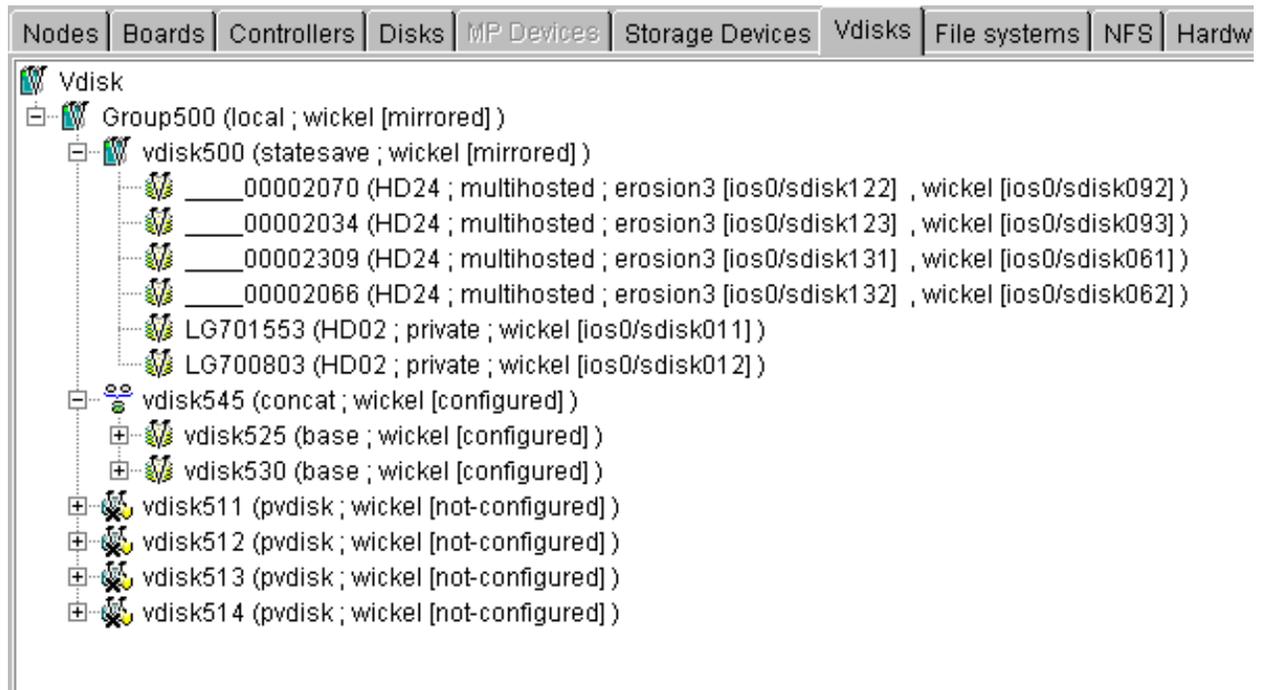


Figure 8: Tree view of "Vdisks" tab (approximates figure)

Sub-tabs

Sub-tabs are provided in order to expand the overview. These sub-tabs add a usage-based summary of vdisks to the display. Sub-tabs are considered here for the usage types *All vdisks*, *Top vdisks*, *Construction vdisks* and *Vdisk groups*.

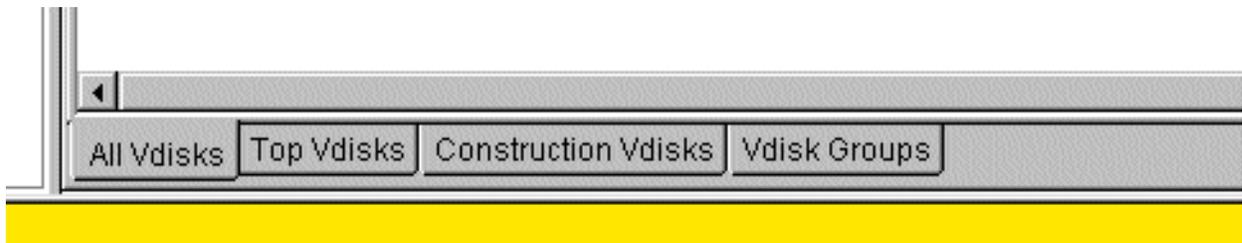


Figure 9: Sub-tab bar

All vdisks

All available vdisks in all groups are displayed.

Top vdisks

The vdisks of the logically highest level are displayed together with their attributes. Only these vdisks can be used for storing data.

If top vdisks are to be used to construct more complex vdisks, you switch following configuration from the *Top vdisks* sub-tab to the *Construction vdisks* sub-tab.

Construction vdisks

Only vdisks of the logically lower levels – i.e. all those below *Top vdisks* – are displayed together with their attributes. These are the vdisks that were used to construct the top vdisks.

Vdisk groups

The vdisk groups are displayed together with their attributes. To facilitate administration of the groups, a vdisk, referred to as a statesave device, is configured for every group under VDisk Lite.

Selection options

Various methods are provided for reducing the number of lines to be displayed in the case of large configurations with several hundred vdisks so that the overview remains clear:

- By using the hierarchy browser (see ...)
- By using the selection configurator (see ...)
- By selecting a restricted sub-tab (see ...)

Administrators can use these methods to filter out the information that is relevant for them.

2.2.2 The VDisk Lite menu structure

When you select the *Vdisks* tab, the following menus and submenus (depending on the context) are offered:

Menu item	Submenu	Function
<i>Create</i>	<i>Create a vdisk group</i>	Creates a new vdisk group
	<i>Create base vdisks (import)</i>	Imports physical disks (no mirroring)
	<i>Create a mirror vdisk</i>	1. Mirrors and imports physical disks 2. Mirrors a maximum of three <i>base</i> vdisks subsequently
	<i>Create a complete sport</i>	1. Mirrors and imports all physical disks for several SCSI ports 2. Mirrors all vdisks for several SCSI ports subsequently
	<i>Create a stripe vdisk</i>	Creates a <i>stripe</i> vdisk
	<i>Create a concat vdisk</i>	Creates a <i>concat</i> vdisk
	<i>Partition a volume</i>	Partitions a vdisk into several <i>P-Vdisks</i>
<i>Modify</i>	□□□□□□□□ --	Makes changes to vdisks
Show old vdisk configuration	□□□□□□□□ --	Shows vdisk configurations that were not created with VDisk Lite
Show the vdisk lite configuration	□□□□□□□□ --	Shows the current VDisk Lite configuration

Table 1: The VDisk Lite menu structure

2.2.3 Object-specific actions

Some of the actions in VDisk Lite can only be performed if certain conditions are fulfilled. These actions are termed object-specific. They will only appear as menu items if at least one line in the table view has been selected **and** the required conditions – for all selected objects assuming several were selected – are fulfilled.

[Table ""](#) lists the object-specific menu items and explains the conditions.

Menu item	Conditions
Export <i>vdisk</i>	1. Can only be used for <i>base</i> or <i>mirror</i> □ <i>vdisks</i> . 2. The vdisk must not be configured. 3. The vdisk must be available (status <i>avail</i> ¹).
Delete <i>vdisks</i>	1. Only possible for <i>stripe</i> or <i>concat</i> □ <i>vdisks</i> . 2. The vdisk must not be configured. 3. The vdisk must be available (status

	<i>avail</i> ¹).
<i>Delete vdisk groups</i>	All top-level vdisks in this group must be available (status <i>avail</i> ¹). Note: The physical disks in this group are also exported here.
<i>Join with next vdisk</i>	The next p-vdisk must be available (status <i>avail</i> ¹).
<i>Delete all partitions</i>	All p-vdisks in this volume must be available (status <i>avail</i> ¹) and must not be configured.
<i>Configure vdisks</i>	The vdisk to be configured must be a top vdisk and must not be configured
<i>Deconfigure vdisks</i>	Only possible if all higher-level vdisks are not configured.
<i>Configure all vdisks in this group</i>	No conditions
<i>Deconfigure all vdisks in this group</i>	No conditions

Table 2: Object-specific actions for vdisks



If vdisks that are not recognized and monitored by *WebSysAdmin* are accessed, an error message is only issued when the command is executed. For example, raw access for an operating system without configuration of the raw partition on the *VConfig* interface.

3 Configuration with VDisk Lite

This chapter describes the typical applications of VDisk Lite. The menus and their input fields are explained and additional information is provided on working with the *WebSysAdmin* user interface.

WebSysAdmin offers a graphical user interface, which you can use to comfortably execute all of the VDisk Lite functions. You will find more detailed information in the manual "[System Administration within a Domain \(Reliant UNIX\)](#)".

3.1 Configuration constraints

Special constraints and limits must be observed when configuring vdisks. In particular, these affect the number and the size of the vdisk pieces and the way they interact with each other.

- Existing vdisk configurations can be displayed with *VConfig*. They can continue to be used without restrictions but cannot be modified with VDisk Lite.



Features such as memory disks and mirroring of root disks are not supported by VDisk Lite. These vdisk types must still be configured with the previous vdisk method. For further details on this topic, refer to the manual "[Virtual Disks](#)".

- Only entire disks can be mirrored. The capacity of the disks must be the same.
- Only disks with synthetic geometry (VDisk Lite is therefore not released for the disk types MP 50, MP65, MP 81 and MP 85.) can be used.
- A virtual disk cannot be defined as a piece of a statesave device.
- Mirror* vdisks can contain at most three mirror pieces.
- Stripe* vdisks cannot be placed over *stripe* or *concat* vdisks or *p-vdisks*.
- If you want to mirror *stripe* vdisks, you have to first define the mirror disks and then, based on this, the *stripe* vdisk.
- All pieces of a *stripe* vdisk must be the same size. The size of the individual pieces must in turn be fully divisible by the cluster size.
- Concat* vdisks cannot be placed over *concat* vdisks or *p-vdisks*.
- To create larger capacity vdisks, it is possible to place another *Concat* vdisk over *stripe* vdisks.
- If a vdisk is split into *p-vdisks*, the resulting *p-vdisks* always represent a complete overlapping of the original vdisk. This original vdisk is termed a *volume*.



Apart from *p-vdisks*, all vdisks only use the complete underlying devices. It is therefore immaterial whether a virtual or a physical disk (or their partition 7) is in question.



The data on a virtual disk will be lost if its configuration data is changed. If, for example, a *stripe* vdisk is defined with two pieces and the definition is changed to three pieces, all data on this virtual disk will be lost.

3.2 Limits on vdisks

- The maximum number of vdisks in VDisk Lite is set by default to 5120, the maximum (can be set using the *idtune* command) value that can be set is 9999.
- The maximum size of a virtual disk can theoretically be 264 Bytes = 16 Exabytes. The maximum sizes of the file systems are subject to the following restrictions however:

File system	Maximum size
ufs	4 Gbytes
vdfs	16 Tbytes

- *Concat* and *stripe* vdisks can contain a maximum of 99 pieces. These must be of type *base* or *Mirror*.
- A vdisk can be split into a maximum of 256 *p-vdisks*. This limit also applies if a *p-vdisk* that has already been created is to be split again.
- A statesave device for a group can hold at most 500 entries.
- The maximum size of a vdisk is derived from the maximum limit of 500 entries per group. Each *base* vdisk counts as two entries here. *Mirror* vdisks are counted as follows however:
 - Each mirror disk in the group counts as one entry
 - Each piece of this mirror disk (at least two pieces) counts as one entry

These values give rise to a maximum number of $500/3 = 166$ mirror disks with two pieces each or $500/2 = 250$ *base* vdisks. The maximum size of a vdisk can be calculated if this is multiplied by the capacity per physical disk.

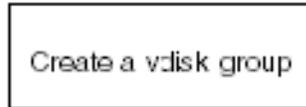
3.3 Configuration procedures

The following sections provide information about the most important dialog boxes for creating the different vdisk types. You will find the necessary information about the input fields as well as details of special features.

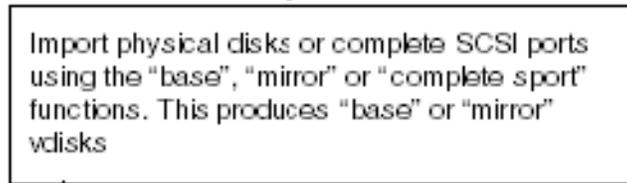
3.3.1 Basic principles

The configuration steps **must** be performed in a fixed sequence. This means that a new (nesting) level is reached with each step, which offers new options. [Figure "Procedure for configuring the various vdisk types"](#) should clarify this:

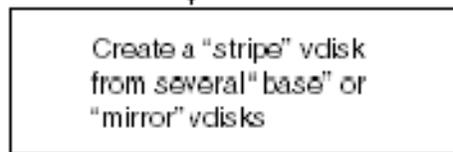
Level 1:
(mandatory)



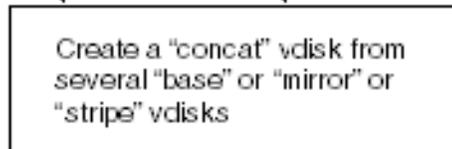
Level 2:
(mandatory)



Level 3:
(optional)



Level 4:
(optional)



Level 5:
(optional)

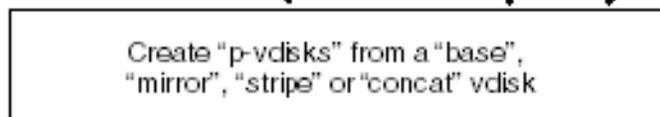


Figure 10: Procedure for configuring the various vdisk types

3.3.2 Creating a new group

Assuming no existing group can be used, a new group must first be created (if no group is available, no vdisks can be created or imported). The following dialog box opens on the screen when you select the menu item Create a vdisk group under Create on the Vdisks tab:

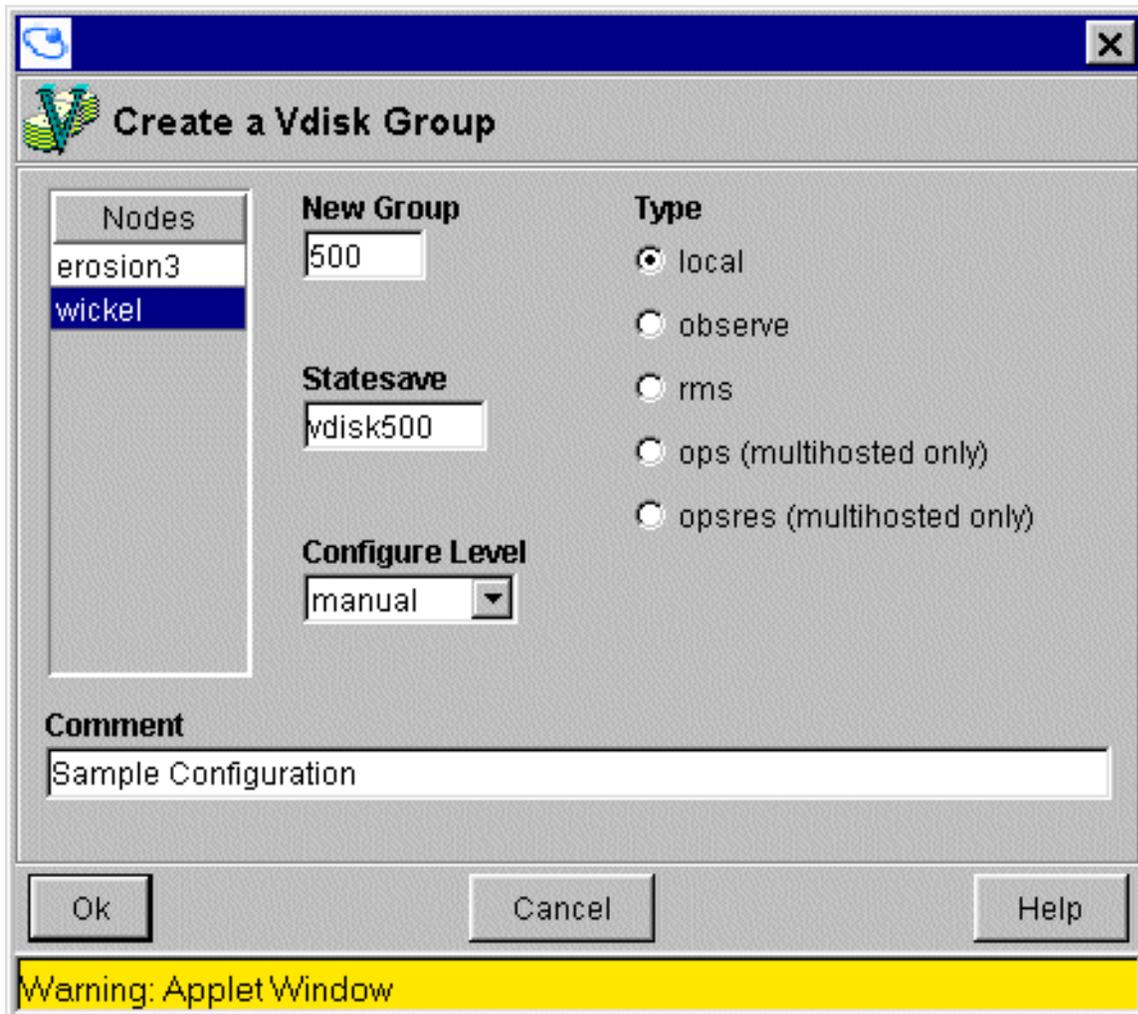


Figure 11: "Create a vdisk group" dialog box (approximates figure)

The dialog box offers the following input options:

Nodes

Select one (computer) node, on which the new group is to be created. You must select at least one node. Multiple selections are also possible, but only make sense if the vdisks in this group are to be created globally.

New group

Input field for the number of the new group name. Only numeric input is permitted. The group must not already exist on the selected node(s).

Type

This is where you can select the group type:

- *local* means that this vdisk group is available on all selected nodes as an independent vdisk group.
- *observe* means that this vdisk group is available globally on the selected nodes as a common vdisk group of this type (here: *observe*).
- *ops*, see *observe*.
- *opsres*, see *observe*.
- *rms*, see *observe*.

Statesave

Input field for the name of the statesave device. An error message is issued if you make an incorrect entry. The conditions for assigning the name are as follows:

- The name must be in the form: **vdisknnnn**
- The name must not have been assigned already on the selected nodes.

Configure level (can only be selected for local groups)

- *boot* means that this vdisk group is to be configured (activated) automatically during the system boot process.
- *manual* means that this vdisk group will only be configured on the instigation of the administrator.



The new vdisk group can only be used after it has been configured.

Comment

Input field for entering a max. 64-character long comment of any sort.

3.3.3 Importing the required vdisks

Each disk planned for use in VDisk Lite must first be imported. The result of this is a vdisk of type *base*. The dialog box for performing the import can be opened by selecting *Create base vdisks (import)* under *Create* on the *Vdisks* tab.



Mirror disks are an exception to this. They can be created directly without first being created as *base* vdisks. You will find the dialog box in question by selecting *Create a mirror vdisk* or *Create a complete sport* under *Create* on the *Vdisks* tab.

Import Disks to Vdisks

Group, Type, Nodes: 500, local, wickel

Produce at: wickel

Configure

Name: vdisk501

Step [num]: 1

Vdisk	Disk	Node	Type
vdisk501	ios0/sdisk093	wickel	HD06
vdisk502	ios0/sdisk062	wickel	HD06
vdisk503	ios0/sdisk092	wickel	HD06
vdisk504	ios0/sdisk094	wickel	HD06
vdisk505	ios0/sdisk061	wickel	HD06
vdisk506	ios0/sdisk063	wickel	HD06
vdisk507	ios0/sdisk091	wickel	HD06
vdisk508	ios0/sdisk090	wickel	HD06

Buttons: Ok, Cancel, Help, Browse

Warning: Applet Window

Figure 12: Dialog box "Import disks to vdisks" (approximates figure)

The dialog box offers the following input options:

Group, Type, Nodes

You can select an existing vdisk group into which the new vdisks being created are to be imported.

Produce at

In the case of global groups containing several nodes, this action allows you to choose the node on which the configuration commands are to be executed.

Configure

If this field is selected, the imported vdisk is configured (activated) after it has been created on the selected node.



You should select the *Configure* field if you want to release the vdisk for use.

Name

Name of the first vdisk that is to be created. The field is preassigned the next valid vdisk name. An error message is issued if you make an incorrect entry. The conditions for assigning the name are as follows:

- The name must be in the form: **vdisknnnn**
- The name must not have been assigned already on the selected nodes.



If VDisk Lite finds names that are already assigned when it is generating the names, a warning is issued and the next free name is then offered.

Step (num)

You can enter an increment here for name generation when simultaneously importing several vdisks.

Browse

You can filter out the relevant disks here.

The disks selected previously in the *AvailDisk Browser* dialog box (see [Figure ""AvailDisk Browser" dialog box with selected disks \(approximates figure\)""](#)) are displayed in table form in the *imports* box.

Ok

The *Ok* button can only be activated when all required entries have been made.

"AvailDisk Browser" dialog box

You will see a listing here of all the disks that meet the specified selection criteria. Several disks can be selected and copied by selecting the *Take it* button (see [Figure "AvailDisk Browser" dialog box with selected disks \(approximates figure\)"](#)) in the table in the *Import disks to vdisks* dialog box (see [Figure "Dialog box "Import disks to vdisks" \(approximates figure\)"](#)).



Where administration domains or systems contain a large number of vdisks or vdisk groups, we recommend you use the selection configurator when making your selection.

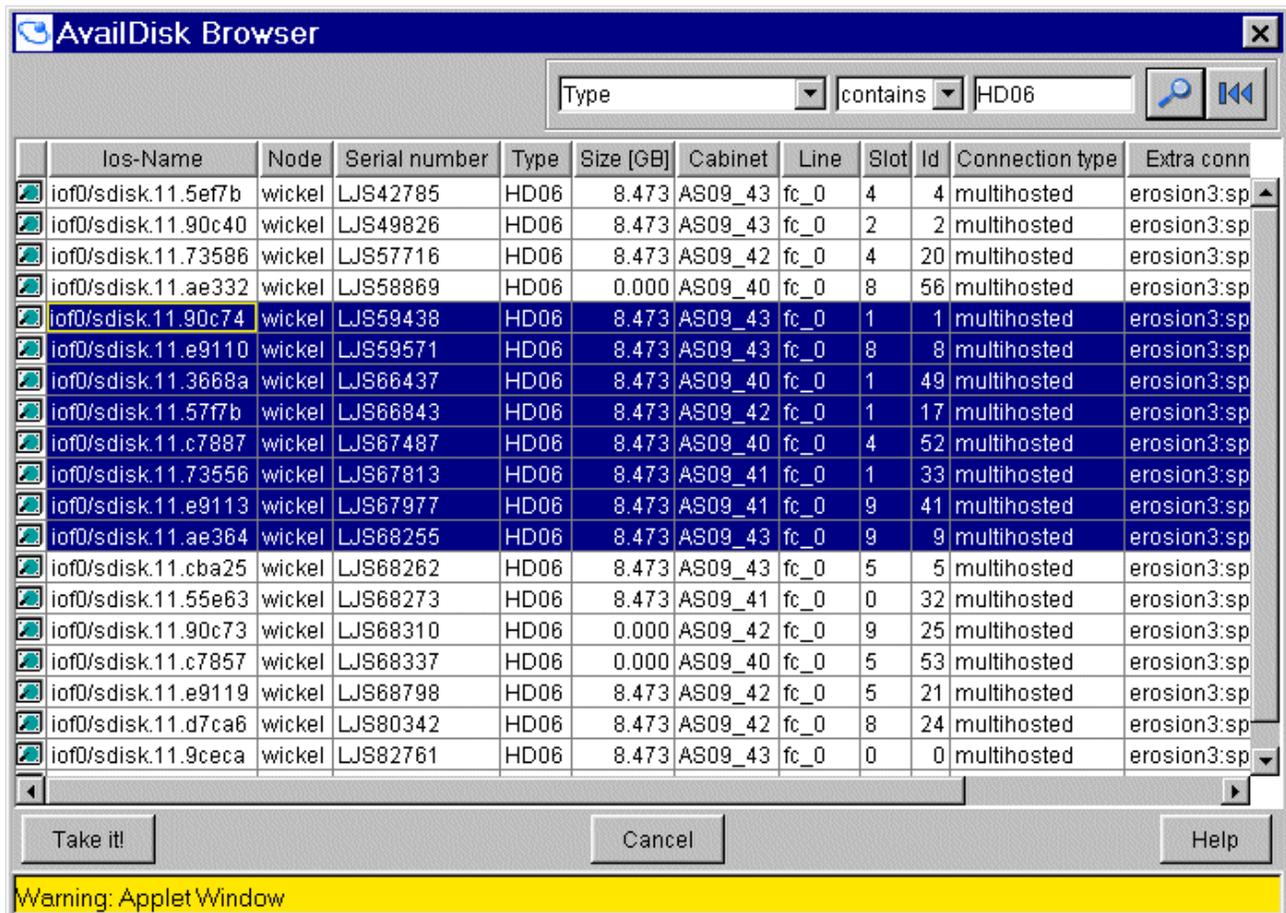


Figure 13: "AvailDisk Browser" dialog box with selected disks (approximates figure)



The *Take it* button can only be activated when a valid name has been entered in the *Name* field in the *Import disks to vdisks* dialog box.

3.3.4 Creating a mirrored vdisk

Mirror vdisks can be created both from physical disks and from imported vdisks. The following dialog box is opened when you select *Create a mirror vdisk* under *Create* on the *Vdisks* tab.

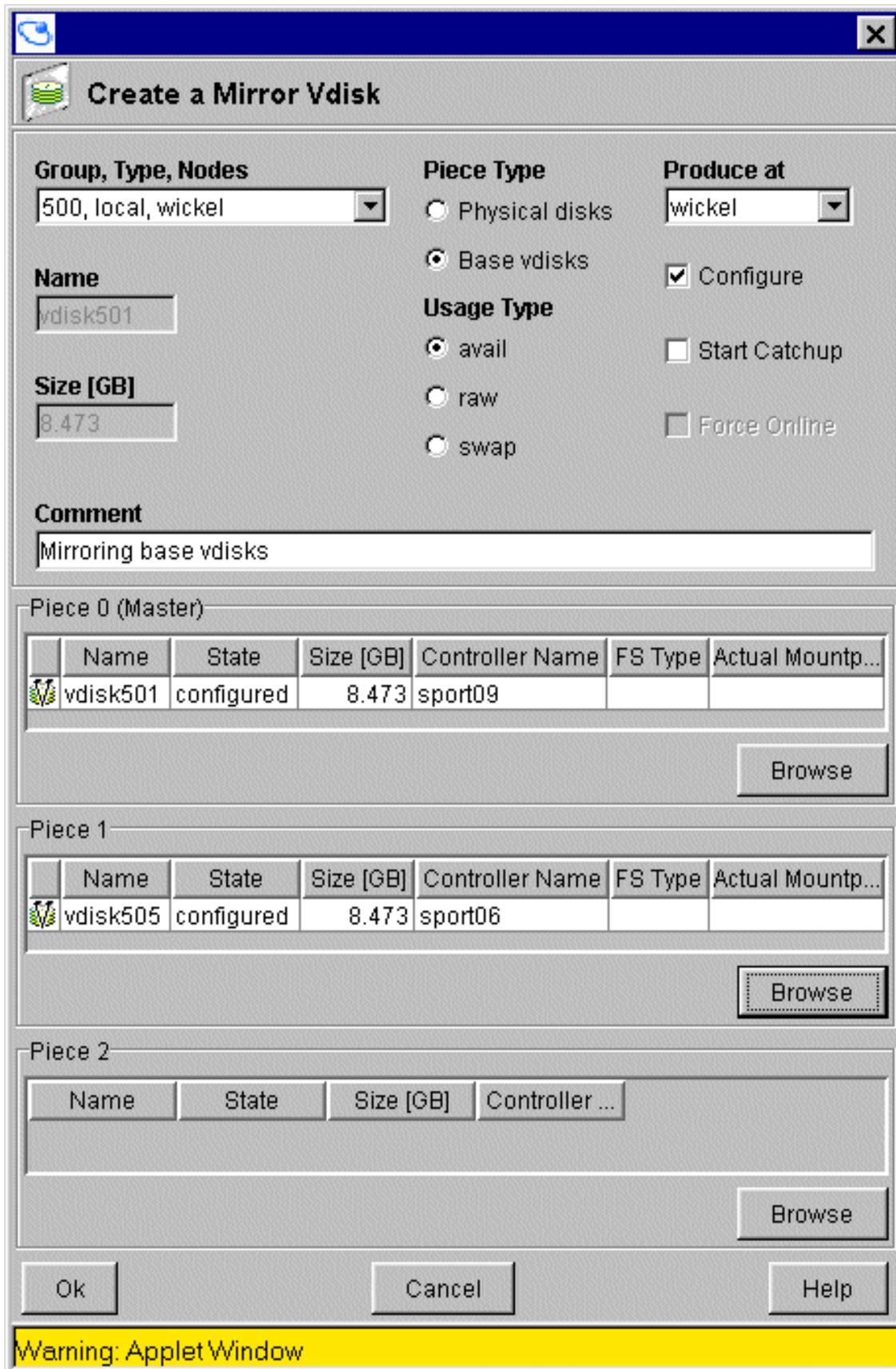


Figure 14: "Create a mirror vdisk" dialog box (approximates figure)

This dialog box offers the following input options:

Group, Type, Nodes

You can select an existing vdisk group here to which the new virtual disk is to belong.

Piece type

Specifies whether *base* vdisks or physical disks should be used.

Produce at

In the case of global groups containing several nodes, this action allows you to choose the node on which the configuration commands are to be executed.

Configure

If this field is selected, the imported vdisk is configured (activated) after it has been created on the selected node.

Name

This is where you enter the name of the vdisk. Your input is dependent on the *Piece type* selection field.



base vdisks: The name of Piece 0 appears.

Physical disks: The name can be assigned freely. An error message is issued if you make an incorrect entry. The conditions for assigning the name are as follows:

- The name must be in the following form: **vdisknnnn**
- The name must not have been assigned already on the selected nodes.



If VDisk Lite finds names that are already assigned when it is generating the names, a warning is issued and the next free name is then offered.

Size (GB)

Specifies the capacity of the vdisk in gigabytes.

Usage type

Specifies whether the vdisk is to be used directly as a raw device (*raw*) or as a swap device (*swap*). If the vdisk is to be used as a piece of other vdisks or later for a file system (*avail*), **avail** must be selected here.

Start catchup

If this field is selected, the catchup process is started with Piece 0 being set to online and the other pieces to *enabled*.



The catchup process can only be started if the vdisk was configured beforehand.

Force online

Online status can be enforced when importing physical disks provided there is no data as yet on the vdisk.

Comment

Input field where you can enter any comment.

Browse (Area: Piece 0)

The system establishes the vdisks or physical disks to be used as *Piece 0* (master piece). Only one line may be evaluated and entered in the selection table by choosing the *Take it* button.

Browse (Area: Piece 1)

Only options relevant for Piece 0 are shown. This area only becomes active if Piece 0 was selected.

Browse (Area: Piece 2)

Only options relevant for Piece 0 are shown. This area only becomes active if Piece 1 was selected.



If you want to create a *mirror* vdisk with only two pieces, no selection must be made in the selection table for Piece 2.

3.3.5 Creating mirrored vdisks for a SCSI port

If you want to mirror all drives in a SCSI port (*sport*) – for reasons of data security for example – you can use the *Create a complete sport* menu item. The following dialog box is opened when you select *Create a complete sport* under *Create* on the *Vdisks* tab:

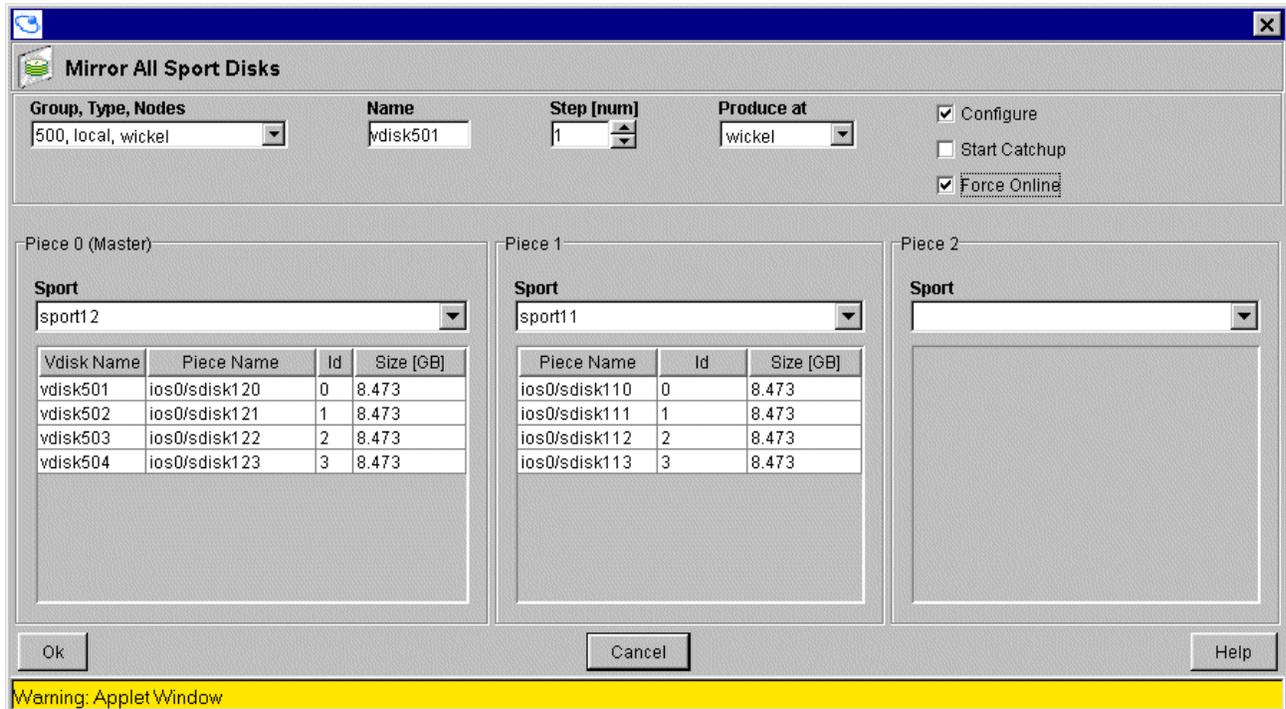


Figure 15: Dialog box "Mirror all sport disks " (approximates figure)

The dialog box offers the following input options:

Group, Type, Nodes

You can select an existing vdisk group here to which the new virtual disk is to belong.

Name

This is where you enter the name of the vdisk. The name can be assigned freely. An error message is issued if you make an incorrect entry. The conditions for assigning the name are as follows:

- The name must be in the following form: **vdisknnnn**
- The name must not have been assigned already on the selected nodes.



If VDisk Lite finds names that are already assigned when it is generating the names, a warning is issued and the next free name is then offered.

Step (num)

You can enter an increment here for name generation when simultaneously importing several vdisks.

Produce at

In the case of global groups containing several nodes, this action allows you to choose the node on which the configuration commands are to be executed.

Configure

If this field is selected, the imported vdisk will be configured after it has been created on the selected node.

Start catchup

If this field is selected, the catchup process is started with Piece 0 being automatically set to online and the other pieces to *enabled*.

The catchup process may take some time depending on the size of the disk. During this time, the progress is indicated as a percentage in the table view in the *Status* field.



The catchup process can only be started if the vdisks were configured beforehand.

Force online

When you select this field, all pieces are set to online, i.e. as if they were already mirrored. This makes sense for new disks that are to be mirrored, provided there is no data as yet on the disks in question.

3.3.6 Creating a stripe vdisk

Choose the menu item *Create a stripe vdisk* under *Create* on the *Vdisks* tab. The following dialog box is opened:

Create a Stripe Vdisk

Group, Type, Nodes
500, local, wickel

Name
vdisk510

Size [GB]
33.892

Comment
Assembling mirror vdisks to a striped volume

Pieces

	Name	State	Size [GB]	Controller Name
	vdisk501	not-mirrored	8.473	
	vdisk502	not-mirrored	8.473	
	vdisk503	not-mirrored	8.473	
	vdisk504	not-mirrored	8.473	

Warning: Applet Window

Figure 16: "Create a stripe vdisk" (approximates figure)

This dialog box offers the following input options:

Group, Type, Nodes

You can select an existing vdisk group here to which the new vdisk is to belong.

Piece type

You can specify the vdisk type of the source disk, choosing between *base* or *mirror*. The RAID level of the vdisk is established here:

- *base* corresponds to RAID level 0
- *mirror* corresponds to RAID level 1+0

Produce at

In the case of global groups containing several nodes, this action allows you to choose the node on which the configuration commands are to be executed.

Configure

If this field is selected, the imported vdisk is configured after it has been created on the selected node.

Name

Name of the first vdisk that is to be created. The field is preassigned the next valid vdisk name. An error message is issued if you make an incorrect entry. The conditions for assigning the name are as follows:

- The name must be in the form: **vdisknnnn**
- The name must not have been assigned already on the selected nodes.



If VDisk Lite finds names that are already assigned when it is generating the names, a warning is issued and the next free name is then offered.

Domain size

You can enter the cluster size in 512-byte blocks. Possible cluster sizes are: 32, 64, 128 and 256 blocks (16, 32, 64 and 128 Kbytes).



Choosing the best individual cluster size can further enhance performance.

Usage type

You specify whether the vdisk is to be used as a raw device (*raw*). In all other cases, *avail* remains selected. The vdisk can continue to be used as a construction vdisk in this way (for creating higher-level vdisks, see [Figure "Procedure for configuring the various vdisk types"](#)) or it can accommodate a file system.

Comment

Input field where you can enter any comment.

Browse (Area: Pieces)

The system establishes the vdisks in question. At least two pieces must be selected and entered in the selection table by choosing the *Take it* button.

3.3.7 Creating a concat vdisk

Choose the menu item *Create a concat vdisk* under *Create* on the *Vdisks* tab. The following dialog box is opened:

Create a Concat Vdisk

Group, Type, Nodes
500, local, wickel

Name
vdisk545

Size [GB]
8.47

Comment
Assembling two base vdisks to one concat vdisk

Piece Type
 base
 mirror
 stripe

Produce at
wickel

Configure

Usage Type
 avail
 raw

Pieces

Name	State	Size [GB]	Controller Name
vdisk525	configured	4.235	sport01
vdisk530	configured	4.235	sport01

Browse

Ok Cancel Help

Warning: Applet Window

Figure 17: "Create a concat vdisk" dialog box (approximates figure)

This dialog box offers the following input options:

Group, Type, Nodes

You can select an existing vdisk group here to which the new vdisk is to belong.

Piece type

You can specify the vdisk type of the source disk, choosing between *base*, *mirror* and *stripe*.

Produce at

In the case of global groups containing several nodes, this action allows you to choose the node on which the configuration commands are to be executed.

Configure

If this field is selected, the imported vdisk is configured after it has been created on the selected node.

Name

Name of the first vdisk that is to be created. The field is preassigned the next valid vdisk name. An error message is issued if you make an incorrect entry. The conditions for assigning the name are as follows:

- The name must be in the form: **vdisknnnn**
- The name must not have been assigned already on the selected nodes.



If VDisk Lite finds names that are already assigned when it is generating the names, a warning is issued and the next free name is then offered.

Usage type

You specify whether the vdisk is to be used as a raw device (*raw*). In all other cases, *avail* remains selected. The vdisk can continue to be used as a construction vdisk in this way (for creating higher-level vdisks, see [Figure "Procedure for configuring the various vdisk types"](#)) or it can accommodate a file system.

Comment

Input field where you can enter any comment.

Browse (Area: Pieces)

The system establishes the vdisks in question. At least two pieces must be selected and copied to the selection table (from the VDisk Browser) by choosing the *Take it* button.

3.3.8 Partitioning a vdisk into several "P-Vdisks"

Open the dialog box for creating *p-vdisks*. To do this, choose the menu item *Partition a volume* under *Create* on the *Vdisks* tab.

Each dialog box is preceded by explanatory text, which makes it even easier to fill in the fields correctly.

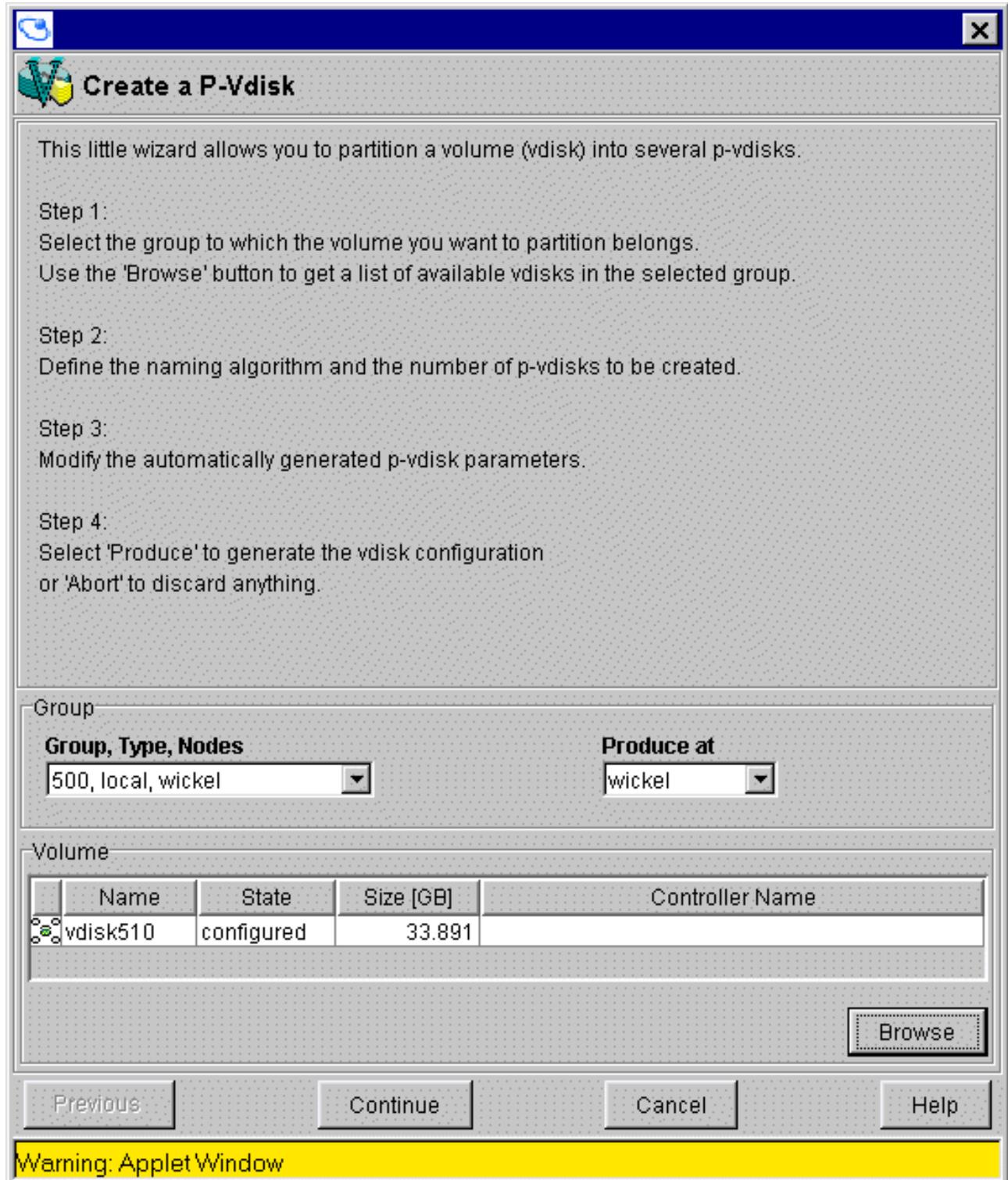


Figure 18: Part 1 of the "Create a p-vdisk" dialog box (approximates figure)

This dialog box offers the following input options:

Group, Type, Nodes

You can select an existing vdisk group here to which the new vdisk is to belong.

Produce at

In the case of global groups containing several nodes, this action allows you to choose the node on which the configuration commands are to be executed.

Browse (Area: Volume)

The system establishes the vdisks to be used. Only one volume can be selected and copied to the selection table (from the AvailDisk Browser) by choosing the *Take it* button.

Click on *Continue*, when you have completed your input.

Part 2 of the dialog box for partitioned vdisks

There are further queries to be answered in the second part of the dialog box.

Create a P-Vdisk

Volume

Volume	Type	Size [GB]	Group
vdisk510	stripe	33.891	500

Enter the number of p-vdisks into which the volume should be divided,

Enter the start value to be used to automatically create the names of the p-vdisks, e.g. if you enter '1230', the first p-disk will get the name 'vdisk1230'.

Enter the step width to be used when automatically creating the vdisk names, e.g. if you enter '10', the second vdisk will be 'vdisk1240', the third 'vdisk1250'...

Enter the default size (no. of 512-byte blocks) of the p-vdisks. This is just the default, you may individually change each size on the next page...

Enter 'yes' if the vdisk configuration is to be activated immediately. Configure

Enter 'raw' if a raw device is to be created

Comment

Previous Continue Cancel Help

Warning: Applet Window

Figure 19: Part 2 of the "Create a p-disk" dialog box (approximates figure)



The size of the **last** p-disk in a volume to be created is derived from the remaining capacity. It cannot be changed. If a **negative value** appears in the *Size* field, the distribution to the individual p-disk must be changed until the minimum size of 2048 blocks (1 Mbyte) has been reached for the last p-disk as well.

Click on *Continue*, when you have completed your input. Click on *Previous*, if you want to switch back to the first part of the dialog box again.

Part 3 of the dialog box for partitioned vdisks

There are further queries to be answered in the second part of the dialog box.

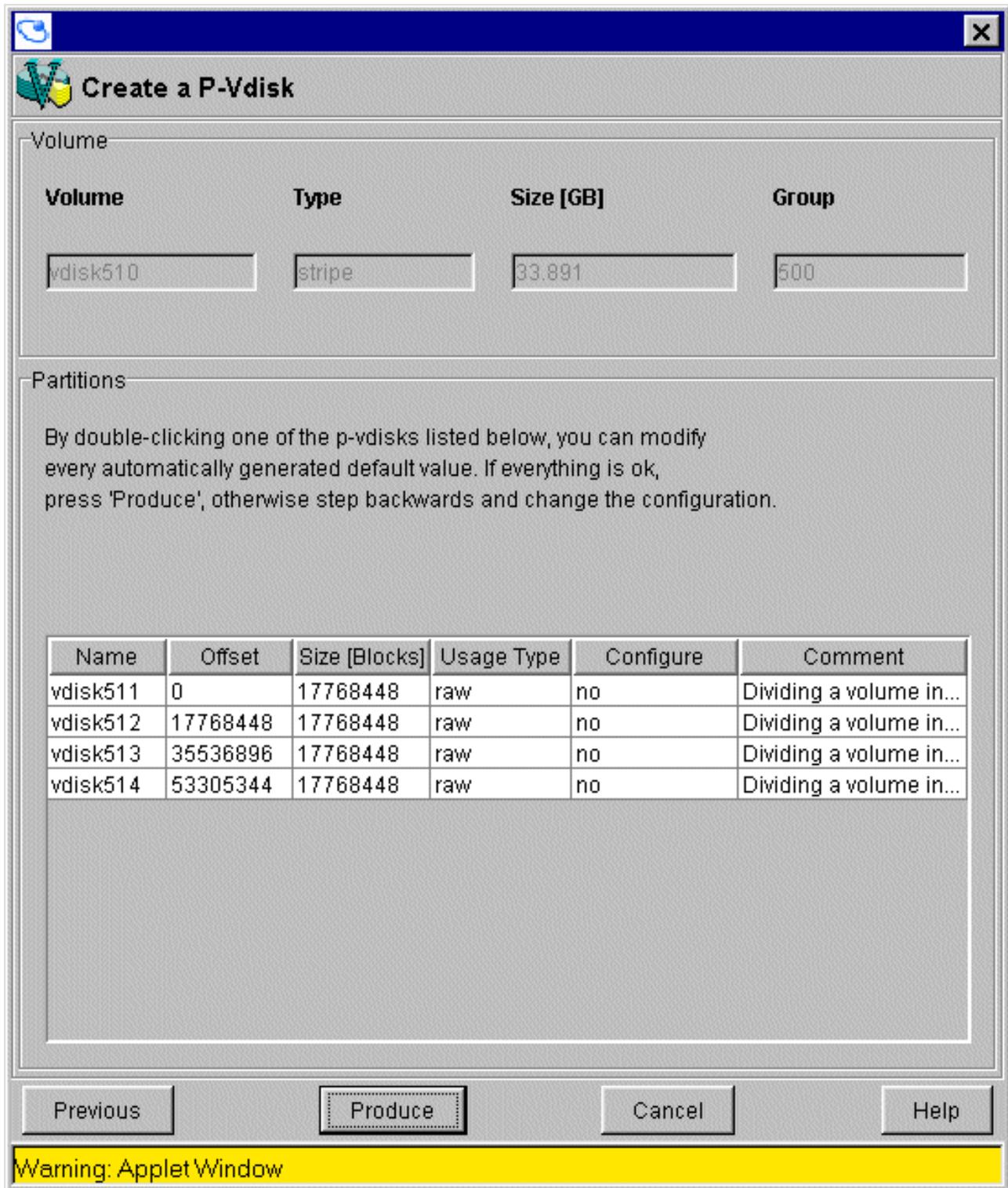


Figure 20: Part 3 of the "Create a p-vdisk" dialog box (approximates figure)

Produce

Starts the process of creating *p-vdisks*. This button can only be activated when all partitions have reached the minimum size set (2048 blocks or 1 Mbyte).

This concludes the dialog boxes. The newly created *p-vdisks* are now displayed in the table view.

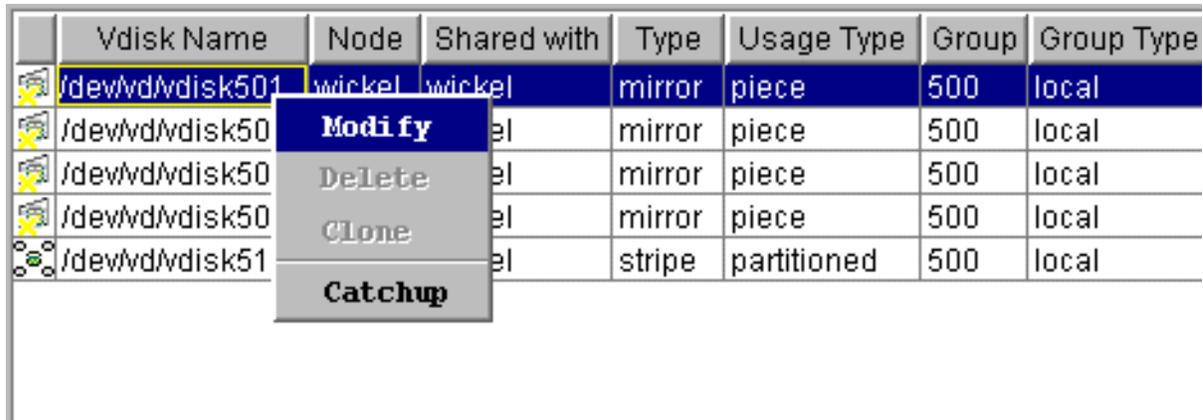
3.3.9 Making changes to vdisks

The attributes of existing vdisks can be changed at any later time. Not all of the actions offered in VDisk Lite are always permitted. Rather it depends more on the current context as to which actions will be possible in each individual case and will therefore be offered as menu items. (See also the [Section "The VDisk Lite menu structure"](#).)

Modifying the attributes of a vdisk

This is how you make changes:

- ▶ First, choose a vdisk from the table view by clicking on the relevant line with the left mouse button. Then, click with the right mouse button on the same line. The following pop-up menu is opened:



	Vdisk Name	Node	Shared with	Type	Usage Type	Group	Group Type
	/dew/vd/vdisk501	wickel	wickel	mirror	piece	500	local
	/dew/vd/vdisk50		el	mirror	piece	500	local
	/dew/vd/vdisk50		el	mirror	piece	500	local
	/dew/vd/vdisk50		el	mirror	piece	500	local
	/dew/vd/vdisk51		el	stripe	partitioned	500	local

The context menu is open over the first row and contains the following items: **Modify**, **Delete**, **Clone**, and **Catchup**.

Figure 21: Opening the pop-up menu for a selected vdisk (approx. figure)

This menu only offers menu items for the currently permitted actions for the selected vdisk. In the figure above, these are the menu items *Modify* and *Catch up*. The *Clone* action is not offered for vdisks. The *Delete* action cannot be executed because of the existing usage as a piece of another vdisk.

- ▶ Now choose *Modify* by clicking on it in order to open the appropriate dialog box. (Instead of clicking with the mouse, you could also choose *Modify* under *Vdisks*.)

Modify Vdisk '/dev/vd/vdisk525' on 'wickel'

Vdisk

Type	Size [GB]	Group
base	4.235	500
FS Type	Mountpoint	Usage Type
		top

Pieces

	iosName	Piece Type	No.	Size [GB]
	/dev/ios0/sdisk012s7	disk	0	4.235

Modify Parameters

Configure

Usage

avail

raw

swap

Comment

Modifying a base vdisk

Ok Cancel Help

Warning: Applet Window

Figure 22: "Modify vdisk..." dialog box (approximates figure)

Each dialog box contains an area called *Modify parameters*. Only the input fields contained within this area

can be modified. In the figure above, these are the following attributes:

Configure

You can change the status of the vdisk here from "configured" to "not configured" or vice versa.



This input field is not shown for *mirror* vdisks.

Usage type

You can change the usage here. Options: *avail*, *raw*, *swap*



- This input field is not shown if the vdisk is being used by a file system.
- The *swap* input field is only shown for *base* and *mirror* vdisks.
- If you click on *raw* or *swap*, the *Configure* input field is selected automatically.

Comment

You can change or add a comment here.

Modifying "mirror" vdisks

Mirror vdisks are a special case in terms of how attributes can be modified. The associated dialog box comprises two tabs, i.e. *Vdisk* and *Change state*.



The *Change state* tab can only be selected if the vdisk is configured.

You can change the following attributes on the *Vdisk* tab:

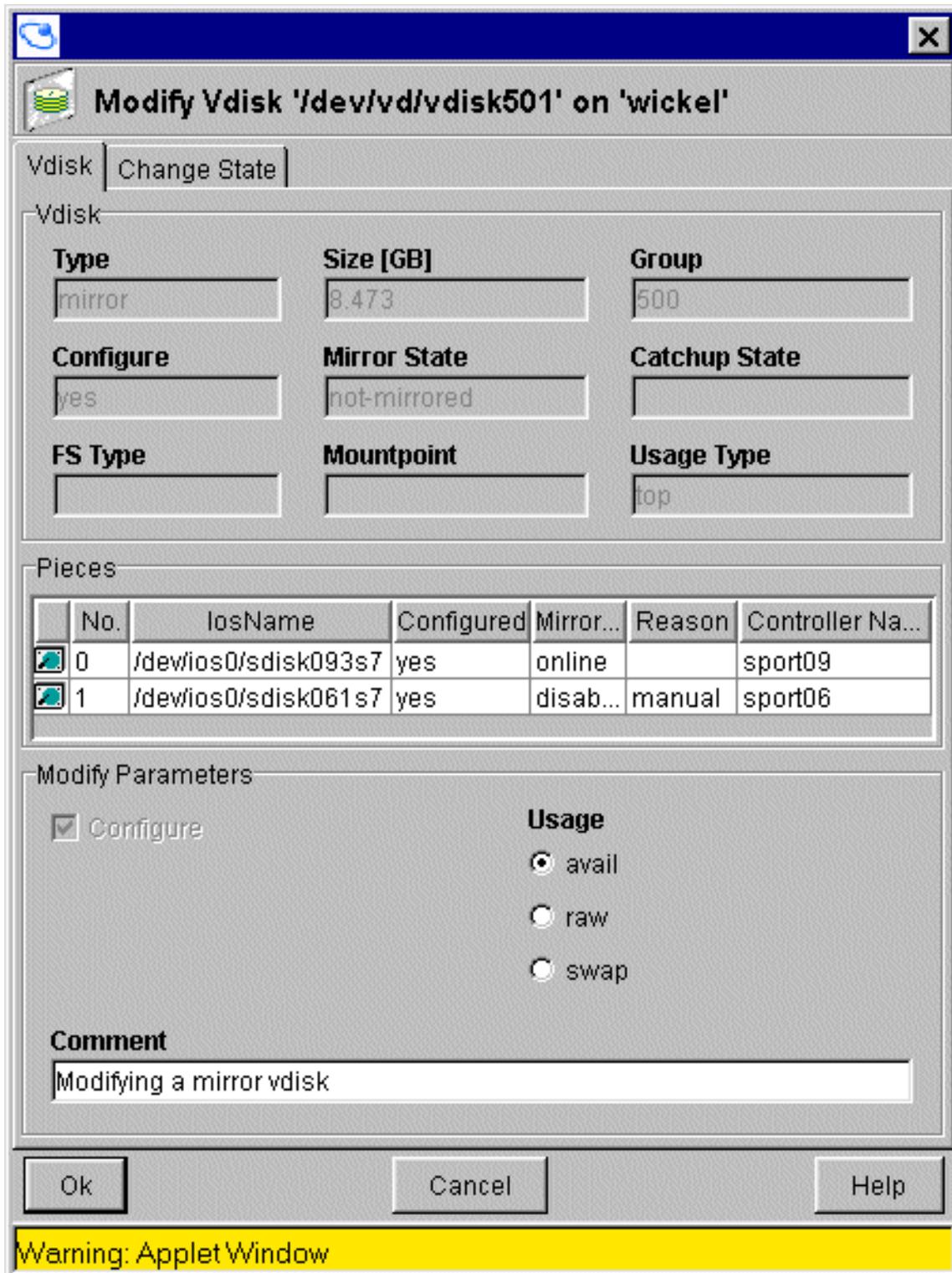


Figure 23: "Modify vdisk ..." dialog box (type "Mirror"), "Vdisk" tab (approx. figure)

Configure

You can change the status of the vdisk from "configured" (selection available) to "not configured" (no selection) or vice versa.

Usage

You can change the usage here.

Comment

You can change the comment here.

You can change the following attributes on the *Change state* tab:

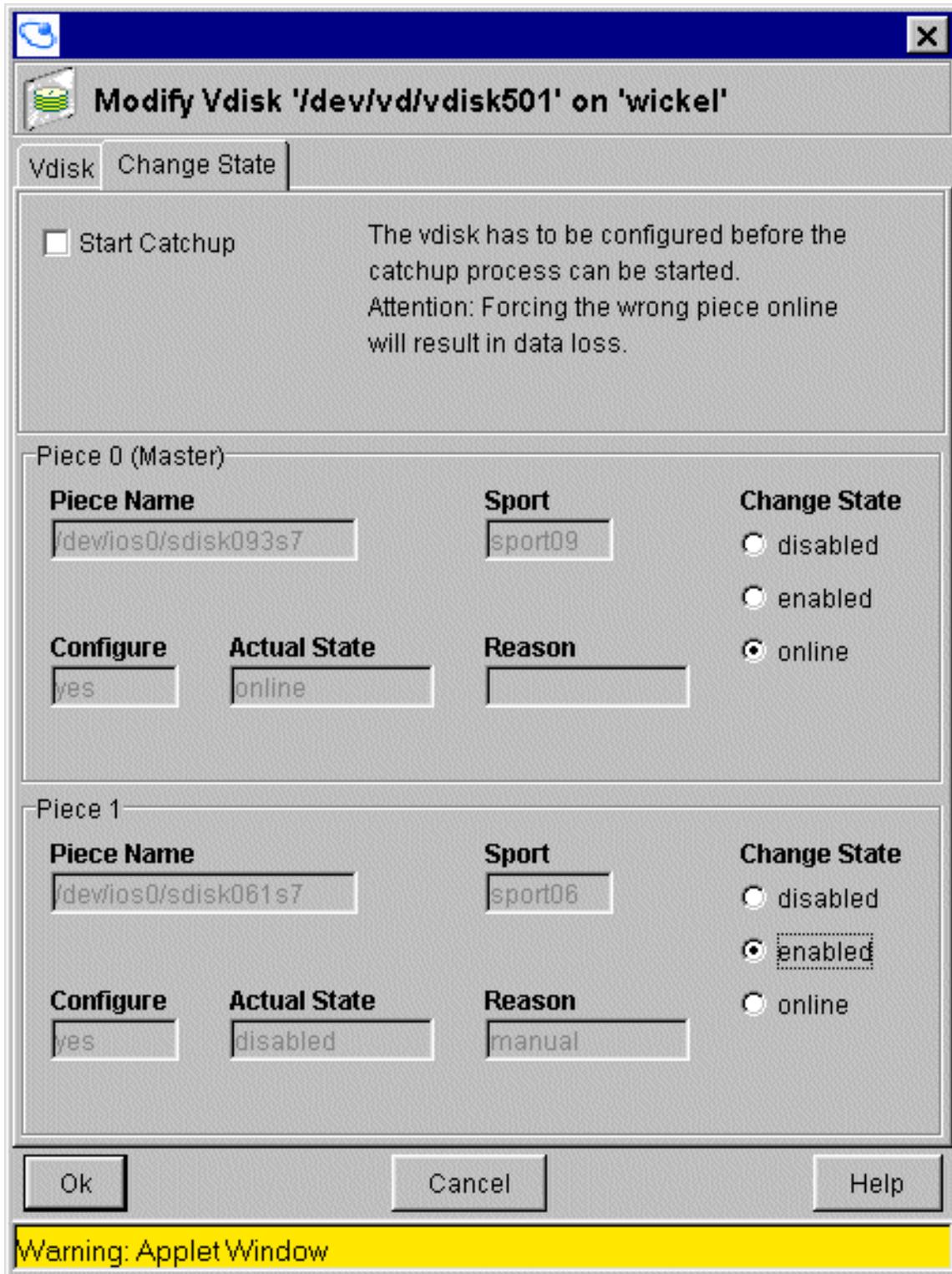


Figure 24: "Modify vdisk ..." dialog box (type "Mirror"), "Change state" tab (approx. figure)

Start catchup

You can manually initiate the catchup process here. If an online piece exists, the catchup process is copied from this online piece to all other pieces on this disk. Otherwise, Piece 0 (default) is set to *online*. All other pieces are set to *enable*. The selections in the input fields for *Change state* are adapted **automatically**.

If the pieces are to be exchanged, use the *Change state* input field in the respective area's *Piece*. Only valid assignments are accepted.



If the status of the vdisk pieces is changed by modifying the individual pieces rather than using the *Start catchup* input field, the configuration commands may not be regarded as meaningful by the operating system.

If a piece with outdated data is selected as the online piece, then all pieces with the old data are overwritten as a result of the manual status change and the starting of a catchup process.

Change state (Piece 0)

You can manually set the status of this mirror disk piece.

Change state (Piece 1)

You can manually set the status of this mirror disk piece.

Modifying the attributes of a vdisk group

You can also modify the attributes of vdisk groups. The following dialog box is opened when you:

- ▶ Step on to the *Vdisk groups* sub-tab.
- ▶ Choose a group from the table view by double-clicking on it. The following dialog box is opened:

Modify Group '500' on 'wickel'

Vdisk

Type
local

Statesave
/dev/vd/vdisk500

Statesave Pieces

	IoName	Controller Name
	/dev/ios0/sdisk093s15	sport09
	/dev/ios0/sdisk062s15	sport06
	/dev/ios0/sdisk092s15	sport09
	/dev/ios0/sdisk061s15	sport06
	/dev/ios0/sdisk012s15	sport01
	/dev/ios0/sdisk011s15	sport01

Modify Parameters

Configure Statesave

Configure Level
manual

Comment
Sample Configuration

Ok Cancel Help

Warning: Applet Window

Figure 25: "Modify group ..." dialog box (approximates figure)

You can modify the following attributes:

Configure statesave

The statesave device is deconfigured in this group by unticking this checkbox.



A group can only be deleted after it has been deconfigured.

Configure level

You can choose between *boot* (automatic configuration at system start) and *manual* (initiated by the system administrator) or vice versa.



This selection is only possible for *local* vdisks.

Comment

You can add a comment here.

4 Configuration example

The basic procedure for configuring vdisks is explained here using a general example. The input required for the default options in the examples is explained briefly for all menus and dialog boxes used.

4.1 The planning phase

In order to work extensively on the disk systems, it is worthwhile to plan the individual steps precisely. This procedure should be clarified if we start with a status analysis and move on to a description of the target configuration and the execution.

4.1.1 Starting situation

Hardware constellation

In our example, four disks are available on each of two SCSI ports (Make sure that sufficient resources are available for providing the necessary disks and installing them symmetrically on two different ports.). Each disk has a capacity of 8 Gbytes. The *autoconf* list shows the following picture:

```
ios0/scon01 53C770 chan:2
ios0/sport06 Id: 8 SE 16 Bit 10MHz
ios0/sdisk063 HD06 -- HD06
ios0/sdisk064 HD06 -- HD06
ios0/sdisk065 HD06 -- HD06
ios0/sdisk066 HD06 -- HD06

ios0/scon02 53C770 chan:1
ios0/sport07 Id: 8 SE 16 Bit 10MHz
ios0/sdisk073 HD06 -- HD06
ios0/sdisk074 HD06 -- HD06
ios0/sdisk075 HD06 -- HD06
ios0/sdisk076 HD06 -- HD06
```

Figure 26: Extracts from the autoconf list



If the capacity of two physical ports is not sufficient, the import procedure can be repeated using additional ports.

4.1.2 Target configuration

- A new group is to be created whose vdisks are to be used for database purposes.
- The disks to be imported are to be operated on two separate SCSI ports.
- The data should be mirrored for security reasons (RAID level 1).
- Due to the high performance requirements, *stripe* vdisks should be used (RAID level 1+0).
- The resulting configuration should be split into several *p-vdisks* of a predetermined size.

Configuration requirements

1. Capacity required: at least 28 Gbytes
2. The distribution among virtual disks is as follows:

Name of vdisk	Required capacity per vdisk	Security requirements	Performance requirements	Function
vdisk51 1	16 Gbytes	mirrored once	striped	DB drive
vdisk51 2	2 Gbytes	mirrored once	striped	DB drive
vdisk51 3	2 Gbytes	mirrored once	striped	DB drive
vdisk51 4	> 8 Gbytes	mirrored once	striped	Reserve vdisk

Table 3: Distribution of required capacity among individual vdisks

Planning the procedure

The following steps must be taken in order to meet the above requirements:

Step 1:

Create a new vdisk group that can accommodate the disks to be imported.

Step 2:

Import a sufficient number of physical disks (here: 4 @ 8 Gbytes) so that the required capacity of >28 Gbytes can be provided. The disks should be mirrored at the same time. Both of these actions can be completed very efficiently in one step using the *Create a complete sport* function.

Step 3:

Combine all four new *mirror* vdisks to a large *stripe* device.

Step 4:

Split the large *stripe* vdisk into several smaller *p-vdisks* of the required size.

Planning step 1:

The new vdisk group must first be created and assigned the group number 500. This new group should then accommodate the new vdisks on a local node.

Planning step 2:

This is where the physical disks that are to be imported and combined to form *mirror* vdisks are selected: Four devices are selected in each case as *Piece 0* (master piece) and *Piece 1* (see tables below).



Additional data security is provided if disks from different SCSI ports are used (in this case *sport06* and *sport07*) for *Piece 0* and *Piece 1*.

Physical device name on sport06	Capacity	Vdisk name	Vdisk type	Capacity
sdisk063	8 Gbytes	vdisk501	mirror (Piece0)	8 Gbytes
sdisk064	8 Gbytes	vdisk502	mirror (Piece0)	8 Gbytes
sdisk065	8 Gbytes	vdisk503	mirror (Piece0)	8 Gbytes
sdisk066	8 Gbytes	vdisk504	mirror (Piece0)	8 Gbytes

Table 4: Overview of disks used for mirroring for the respective Piece 0

Physical device name on sport07	Capacity	Vdisk name	Vdisk type	Capacity
sdisk073	8 Gbytes	vdisk501	mirror (Piece1)	8 Gbytes
sdisk074	8 Gbytes	vdisk502	mirror (Piece1)	8 Gbytes
sdisk075	8 Gbytes	vdisk503	mirror (Piece1)	8 Gbytes
sdisk076	8 Gbytes	vdisk504	mirror (Piece1)	8 Gbytes

Table 5: Overview of disks used for mirroring for the respective Piece 1

The result of these deliberations is four *mirror* vdisks with a capacity of 8 Gbytes each.

Planning step 3:

In order to meet requirements regarding performance and capacity, all of the above *mirror* vdisks created have to be combined in the next step in a large *stripe* vdisk – a so-called *volume*. In our example, this volume will be referred to as *vdisk510*.

Vdisk name	Name of the volume
vdisk501	<input type="checkbox"/> <input type="checkbox"/> vdisk510
vdisk502	
vdisk503	
vdisk504	

Table 6: Combination of mirrored devices in a "stripe" volume

Planning step 4:

It is only at this stage that the required segments can be created in the desired sizes (see table below). *vdisk510* is partitioned here into several smaller *p-vdisks* of the required size:

Required capacity per vdisk	Name of the vdisk
16 Gbytes	vdisk511
2 Gbyte	vdisk512
2 Gbyte	vdisk513
Remainder of partition (12 Gbytes)	vdisk514

Table 7: Distribution of the required capacity to the individual vdisks

The reserve partition (*vdisk514*) is assigned the residual capacity of the 32-Gbyte volume, i.e. is approx. 12 Gbytes large. All requisite database devices (*p-vdisks*) are considered here.

The Section "The implementation phase" shows how the requirements in the example can be fulfilled with the assistance of the user interface.

4.2 The implementation phase

In order to configure the example shown in the previous section, all of the input required in the respective dialog boxes for the individual steps is specified in this section, with comments in some cases.

Step 1: Create a new group

- ▶ Under *Create* in the *Vdisks* menu, choose the menu item *Create a vdisk group*. The *Create a vdisk group* dialog box then opens:

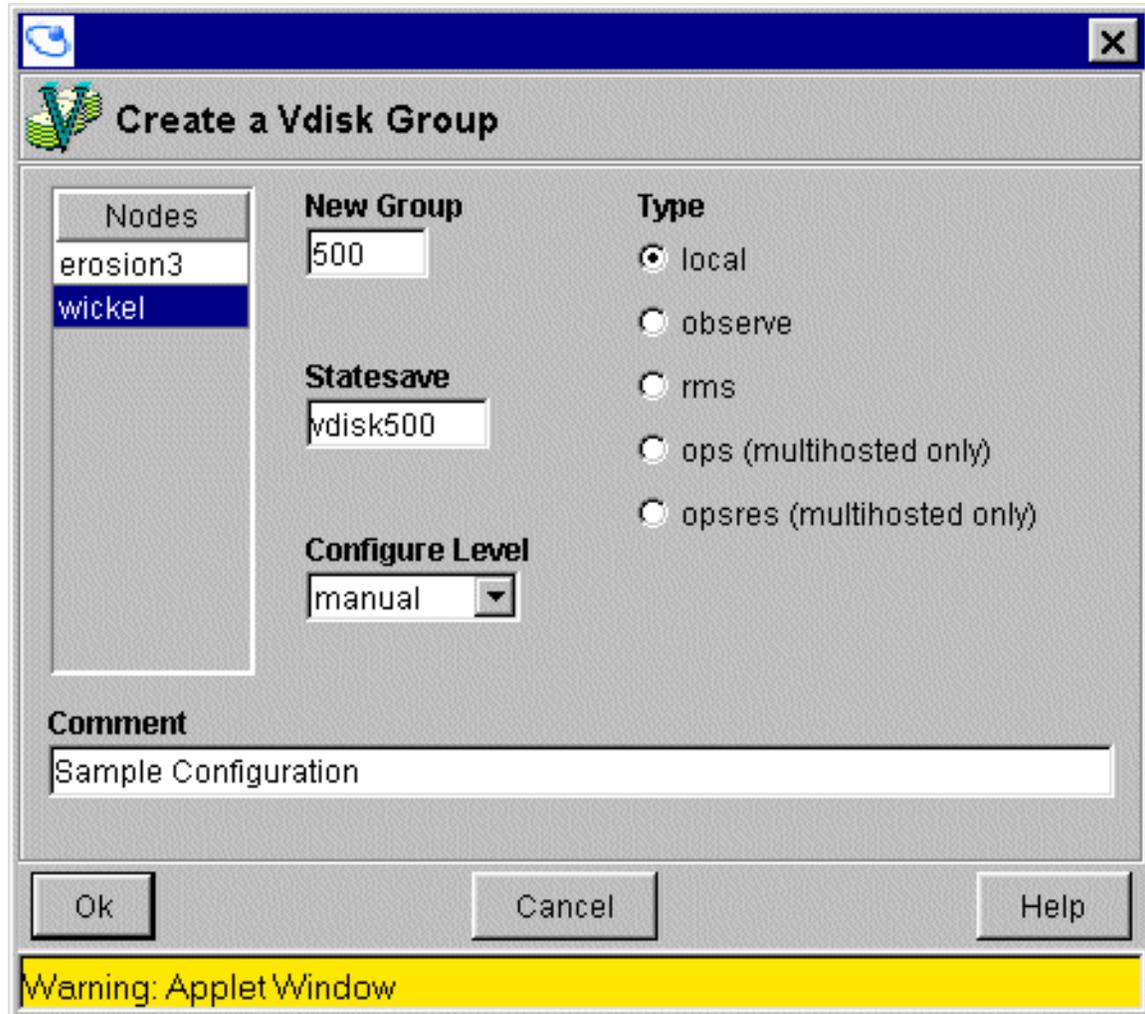


Figure 27: "Create a vdisk group" dialog box

- ▶ In the *Nodes* input field, choose the (computer) node on which the new group is to be created (in the example: *wickel*).
- ▶ Enter the number of the new group in the appropriate field (in the example: *500*).
- ▶ Decide whether this vdisk group is only to available locally (selection button *local*) on one node or globally (selection buttons *observe*, *rms*, *ops*, *opsres*) throughout the entire administration domain (in the example: *local*).
- ▶ Enter the name of the statesave device (in the example: *vdisk500*).
- ▶ In the *Config Level* drop-down menu, choose whether this vdisk group is to be configured (activated)

automatically during the boot process or whether this should be done manually (in the example: *manual*).

- ▶ Enter any comment.
- ▶ Click on *Ok*.

The new group is now created.

Step 2: Import and mirror the required disks

This step can be performed very efficiently.

- ▶ Under *Create* in the *Vdisks* menu, choose *Create a complete sport*. The following dialog box opens:

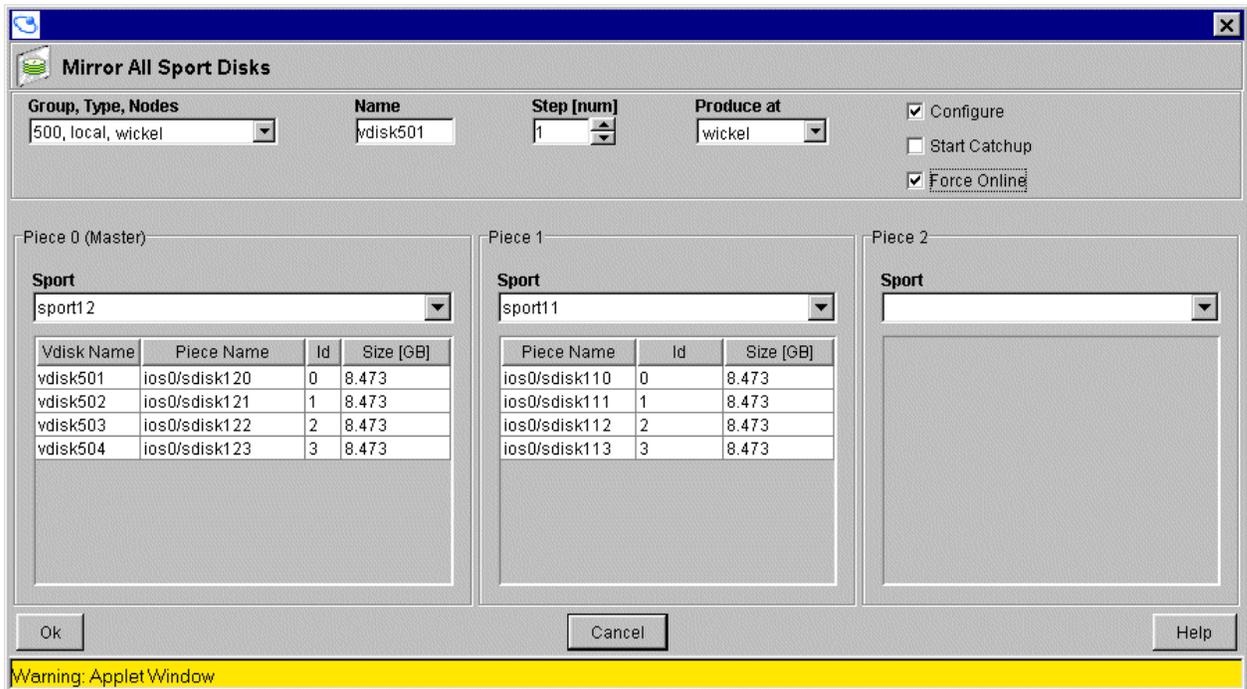


Figure 28: Dialog box for creating mirrored vdisks for a sport

- ▶ In the drop-down menu *Group, Type, Nodes*, choose the group into which your physical disks are to be imported (in the example: *500, local, wickel*).
- ▶ VDisk Lite automatically assigns a name to the first new vdisk (in the example: *vdisk501*).
- ▶ Select the *Configure* field, so that the newly created vdisks can then continue to be used immediately.
- ▶ Select the *Force online* field, because there is no data on the disks shown in the example that could be mirrored.
- ▶ Import and mirror these eight devices in one step by pressing the *Ok* button.

These disks are now available as *mirror* vdisks. The result can be viewed on the *Top vdisks* sub-tab:

vdisk name	State	Storage path	Team	Group	Storage Type	Scale	FS Type	Volume	Capacity	Comment
vdisk501	mirror	wickel	500	local	rd mirror			8.473	8.473	Mirrored vdisk
vdisk502	mirror	wickel	500	local	rd mirror			8.473	8.473	
vdisk503	mirror	wickel	500	local	rd mirror			8.473	8.473	
vdisk504	mirror	wickel	500	local	rd mirror			8.473	8.473	

Figure 29: Result of mirroring: Four "mirror" vdisks

- ▶ Go on to step 3.

Step 3: Create a "stripe" vdisk

From the mirror vdisks you created *vdisk501 - vdisk504*, now create a single large volume, i.e. a logical *stripe* vdisk with a total capacity of 32 Gbytes.

- ▶ Under *Create* in the *Vdisks* menu, choose the *Create a stripe vdisk* menu item. The following dialog box opens:

Create a Stripe Vdisk

Group, Type, Nodes
500, local, wickel

Piece Type
 mirror
 base

Produce at
wickel

Name
vdisk510

Size [GB]
33.892

Domain Size
32

Usage Type
 avail
 raw

Configure

Comment
Assembling mirror vdisks to a striped volume

Pieces

	Name	State	Size [GB]	Controller Name
	vdisk501	not-mirrored	8.473	
	vdisk502	not-mirrored	8.473	
	vdisk503	not-mirrored	8.473	
	vdisk504	not-mirrored	8.473	

Browse

Ok **Cancel** **Help**

Warning: Applet Window

Figure 30: "Create a stripe vdisk" dialog box

- ▶ Choose the group *500* in the drop-down menu *Group, Type, Nodes*.
- ▶ Assign the name *vdisk510* to the vdisk you are creating.
- ▶ Under *Piece type*, enter the vdisk type of the source disk (in this case: *mirror*).
- ▶ Have the system search for the relevant vdisks (Browse button).
- ▶ In the *Usage type* input field, enter *avail* as the usage type to enable continued use as a construction vdisk.
- ▶ Enter the cluster size in the *Cluster size* field.

- ▶ Use this opportunity to enter a meaningful comment.
- ▶ Confirm your selection by clicking on *OK*.

You have now created a virtual device with the name *vdisk510*. This type of vdisk is termed a *volume*. The following figure shows how the result may look.

vdisk name	Label	Share path	Type	Group	Pool Type	Size	FS Type	Serial Number	Block Size	Comment
vdisk510			striped		local	16 Gbytes			1024	

Figure 31: The result of "striping": A vdisk with the name "vdisk510"

Step 4: Partition the volume into the required p-vdisks

You now split this *vdisk510* into partitions of the required size. The following table shows the details.

Logical device name of volume	Capacity of volume	Name of new p-vdisks	Vdisk type	Required capacity
<input type="checkbox"/> <input type="checkbox"/> vdisk510	<input type="checkbox"/> <input type="checkbox"/> 32 Gbytes	vdisk511	p-vdisk	16 Gbytes
		vdisk512	p-vdisk	2 Gbytes
		vdisk513	p-vdisk	2 Gbytes
		vdisk514	p-vdisk	12 Gbytes

Table 8: Overview of required p-vdisk sizes

- ▶ Invoke the dialog box for creating p-vdisks (see ...). Under *Create* in the *Vdisks* menu, choose the *Partition a volume* menu item and follow the instructions in the dialog box.

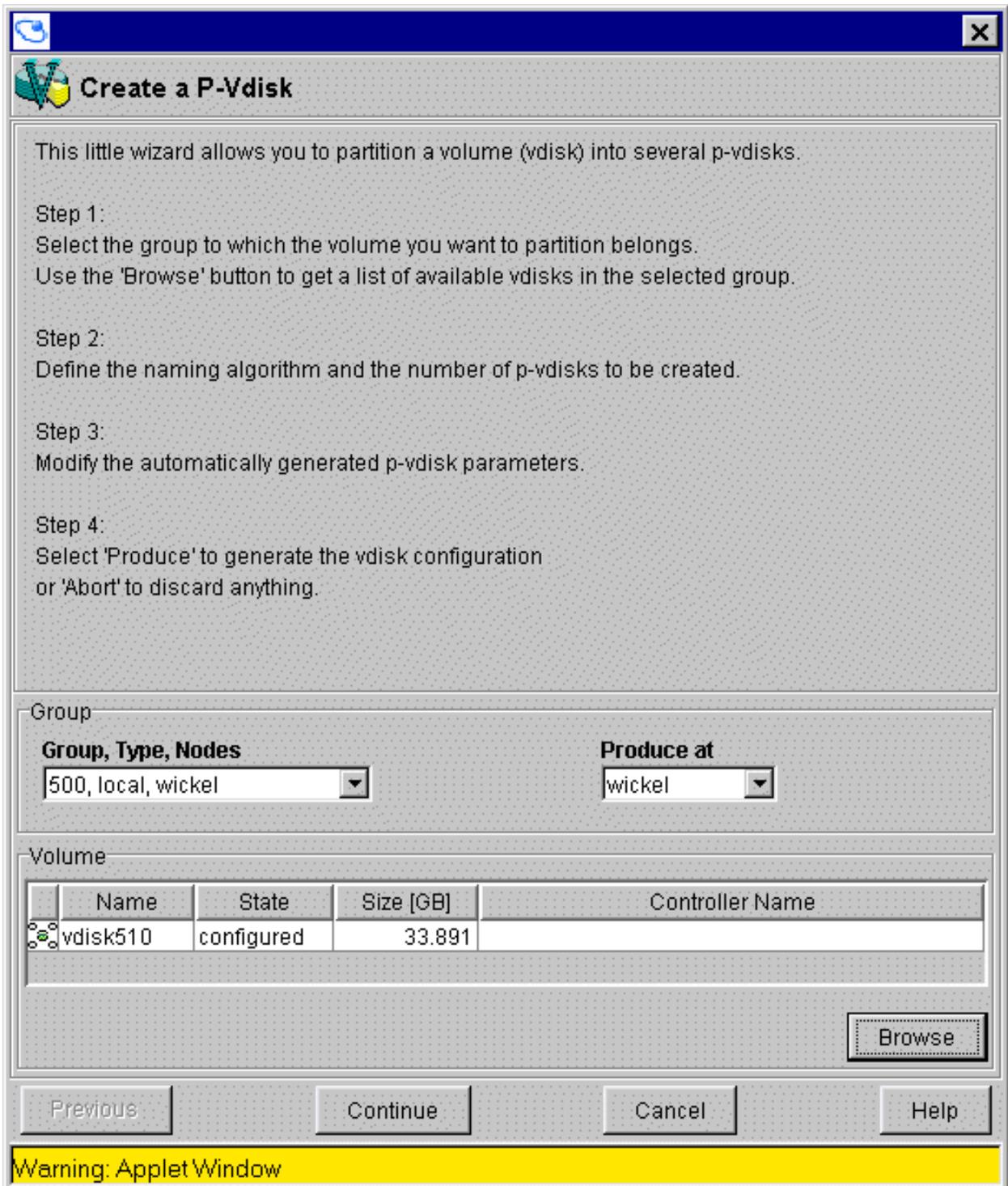


Figure 32: Part 1 of the "Create a p-vdisk" dialog box

- ▶ Choose the group, type and node (in the example: *500, local, wickel*).
- ▶ Have the system search for the relevant vdisks. To do this, choose the *Browse* button.
- ▶ Now choose one of the vdisks offered from the VDisk Browser (in the example *vdisk510*).

Create a P-Vdisk

Volume

Volume	Type	Size [GB]	Group
vdisk510	stripe	33.891	500

Enter the number of p-vdisks into which the volume should be divided,

Enter the start value to be used to automatically create the names of the p-vdisks, e.g. if you enter '1 230', the first p-vdisk will get the name 'vdisk1 230'.

Enter the step width to be used when automatically creating the vdisk names, e.g. if you enter '10', the second vdisk will be 'vdisk1 240', the third 'vdisk1 250'...

Enter the default size (no. of 512-byte blocks) of the p-vdisks. This is just the default, you may individually change each size on the next page...

Enter 'yes' if the vdisk configuration is to be activated immediately. Configure

Enter 'raw' if a raw device is to be created

Comment

Previous Continue Cancel Help

Warning: Applet Window

Figure 33: Part 2 of the "Create a p-vdisk" dialog box

- ▶ Now fill in all the fields.
- ▶ When you have made and checked all your entries, step on to the next dialog box by choosing *Continue*.

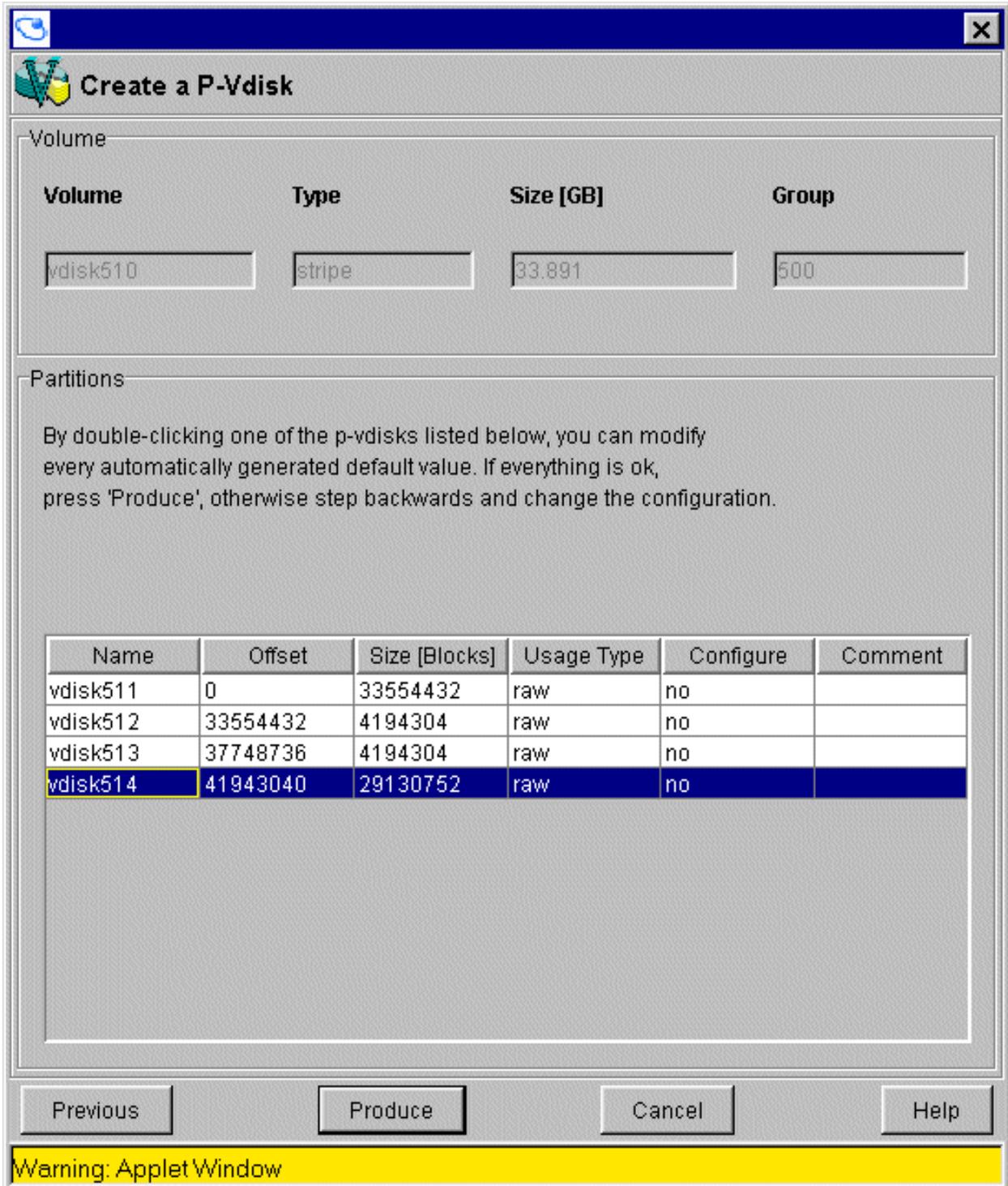


Figure 34: Part 3 of the "Create a p-vdisk" dialog box

The new *p-vdisks* to be created are listed initially with their default values (all partitions are the same size) within the *Partitions* area of the dialog box.

- ▶ Adjust the values as required in line with the description given (*Partitions area*) in the previous dialog box (in the example: 16, 2, 2 and >8 Gbytes). To do this, double-click on the individual table entries.



The size of the **last** p-vdisk in a volume to be created is derived from the remaining capacity. It cannot be changed. If a **negative value** appears in the *Size* field, the distribution to the individual p-vdisks must be changed until the minimum size of 2048 blocks (1 Mbyte) has been reached for the last p-vdisk as well.

▶ Click on *Produce* to terminate the procedure.

The reserve *vdisk514* partition is assigned the remainder of the 32-Gbyte large volume, i.e. is approx 12 Gbytes large. All of the requisite database devices in the target configuration have now been created. The following diagram shows how the results may look.

	Vdisk Name	Node	Shared with	Type	Group	Group Type	State	FS Type	Actual Mountpoint	Size [GB]	Comment
	/dev/vd/vdisk511	wickel	wickel	pvdisk	500	local	not-configured	raw		16.000	
	/dev/vd/vdisk512	wickel	wickel	pvdisk	500	local	not-configured	raw		2.000	
	/dev/vd/vdisk513	wickel	wickel	pvdisk	500	local	not-configured	raw		2.000	
	/dev/vd/vdisk514	wickel	wickel	pvdisk	500	local	not-configured	raw		13.891	

Figure 35: The result of the partitioning process: Four p-vdisks with a total capacity of 32 Gbytes

That completes the configuration example.

Glossary

base vdisks

Base vdisks are the most basic of all vdisks, i.e. all other vdisk types are based on this vdisk, for example *Mirror*, *Stripe*, *Concat* and *P-Vdisk*.

catchup (process)

A catchup process is used with *mirror disks* to synchronize the two pieces of the mirror disk. When the catchup process has ended, the mirror pieces contain the same data.

complex vdisks

Vdisks that comprise simple vdisks of type *Base* or *Mirror*.

concat vdisks

A virtual *Concat* disk is a concatenation of two or more disks. Using virtual disks of this type gives rise to disks whose size is limited exclusively by the maximum size of the file system.

construction vdisks

A vdisk that can be used to create higher-level vdisks. Such a vdisk must be set to the status *avail*.

cylinder

All tracks of a magnetic disk that have the same radius but can lie on different surfaces are referred to as cylinders. All tracks of a cylinder can be read without the write/read head having to be repositioned.

device number

Identical in the case of virtual disks with the *unit number*.

disabled

Possible status of a piece of a *mirror disk*. The piece of the mirror disk with the status *disabled* cannot be accessed for input or output purposes. All newly configured pieces of mirror disks are set to the status *disabled*. Pieces for which I/O access failed are automatically switched to this state by the system.

DOWN

Possible status of a *mirror disk*. If no piece of a *mirror disk* is set to *online*, the mirror disk can no longer be accessed and consequently has the status *DOWN*.

enabled

Possible status of a piece of a *mirror disk*. A piece with the status *enabled* cannot be used by users or programs for I/O access. However, the piece is available as the destination of a *catchup process*. Following a successful catchup process, the piece of the mirror disk goes *online*.

exporting vdisks

When a vdisk is exported, the virtual disk becomes a physical disk again, which can no longer be used within VDisk Lite.

hot swap

In the case of hot swap enabled SCSI tiers (in RM600-E, RM400-Cxx and in the BG70/BG71 or BG31, BG32 and BG42 auxiliary cabinets), disks can be replaced online even if other disks are still active.

import

Physical disks must be imported first before they can be used by VDisk Lite.

master piece

The piece of a virtual mirror disk that by definition contains valid data. The master piece is used as a source of data in the catchup process. This is Piece 0 (default) in the case of VDisk Lite.

mirror disks

Mirror disks are virtual disks that execute all I/O operations simultaneously on two or more disks. The pieces of a *mirror disk* can themselves be either physical or virtual disks.

When you write to mirror disks, the new data is basically written to all pieces. The write speed of a mirror disk is therefore as slow as its slowest piece.

When reading from mirror disks, the data is read from the piece that can make it available the fastest. A *mirror disk* can therefore supply the required data faster than a single disk.

Mirror disks considerably increase the security and availability of disks and therefore the entire system. I/O access to mirror disks can be performed without restrictions as long as at least one piece of the mirror disk is running correctly. If pieces of mirror disks fail, they can generally be repaired without the system having to be brought down first. Mirror disks are the only disks that can also be used for the root and swap area.

mirror vdisk

See *mirror disk*.

MIRRORED

Possible status of a *mirror disk*. A mirror disk is set to *MIRRORED* if at least two pieces of the mirror disk are set to *online*.

NOT-MIRRORED

Possible status of a *mirror disk*. A mirror disk is set to *NOT-MIRRORED* if only **one** piece of the mirror disk is set to *online*. The mirror disk can continue to be used without problems. However, the added security of the data is no longer guaranteed.

offset

An offset is a displacement in relation to a starting point. An offset can be specified for virtual disks in order to define the start block for the virtual disk within a physical *partition*. An offset specifies the first block of the virtual disk in the physical *partition*.

online

Possible status of a piece of a *mirror disk*. All write access to the mirror disk is performed on a piece with the status *online*. Read access to the mirror disk is only ever performed on the *online* piece, which

provides the data the fastest.

online replacement

Replacing a disk when the system is online.

partition

A partition is a piece of a physical disk.

piece

See *piece of a virtual disk*.

piece of a virtual disk

A virtual disk can, depending on its type, comprise one, two or more pieces. Thus, for example, *Base* vdisks always comprise only one piece. *Mirror disks*, on the other hand, must always comprise two pieces.

pop-up menu

Menu that only appears in accordance with the current environmental conditions. For example, a complex vdisk cannot be exported because it is composed of other vdisks. Consequently, no menu is displayed for it.

p-vdisks

A *P-Vdisk* is a segment of a *volume* created by partitioning.

raw disk area

A raw disk area is an area that can be accessed directly by one or more applications without using a file system. The disks in this case can be physical disk pieces or virtual disks.

root mirror disk

The only type of virtual disk that can also be used in the root or primary swap area of the system. Root *mirror disks* are created with the same commands and managed like normal mirror disks. However, since normal virtual disks require information from the root area, it is a more complicated task to create a root mirror disk.

sport (SCSI port)

sport refers to a SCSI controller channel.

statesave device

See *virtual statesave disks*.

top vdisks

A vdisk that is not part of a more complex vdisk. Rather, it can contain other vdisks (unless it is a base vdisk). An administrator can only work with this type of vdisk. The opposite to a top vdisk is an *internal* vdisk.

unit number

Every virtual disk is assigned a number, referred to as a unit number, so that a system can support a number of virtual disks. The unit number is derived from the device entry for the virtual disk (*/dev/vd/vdisk8*). The numbering begins with zero and is set to a fixed limit of 9999.

virtual statesave disks (statesave device)

A *virtual statesave disk* is used for non-volatile storage of the vdisk status. This data requires relatively little space. A statesave vdisk can therefore be kept small (only a few kilobytes). These disks cannot be used by users or application programs. A *statesave device* can contain up to 500 entries.

volume

A volume is a *Base*, *Mirror*, *Stripe* or *Concat* vdisk that can be partitioned into several *P-Vdisks*.

Abbreviations

DAPS	DATA Pool Service
Ebyte	Exabyte
FS	File System
Gbyte	Gigabyte
GUI	Graphical User Interface
HW	Hardware
ID	Identifier
Kbyte	Kilobyte
Mbyte	Megabyte
MTBF	Mean Time Between Failure
OLR	Online Replacement
OPS	Oracle Parallel Server
Pbyte	Petabyte

RAID	Redundant Array of Independent Disks
RMS	Reliant Monitor Software
SCSI	Small Computer Systems Interface
S-Port	SCSI Port (of a SCSI controller)
Tbyte	Terabyte
UFS	UNIX File System
UI	User Interface
Vdisk	Virtual Disk
VxFS	Veritas Extended File System

Related publications

Ordering manuals

For ordering manuals, please contact your local Fujitsu Siemens office. Many manuals are also available on the Web at: □

http://www.siemens.de/servers/man/man_us.htm

Siemens AG publications

Reliant UNIX 5.44

Virtual Disks

System Administrator Guide

Target Group

System administrators

Contents

Describes the different types of virtual disks and how they are configured.

Reliant UNIX 5.45

Reliant UNIX Installation

System Administrator Guide

Target Group

System administrators

Contents

Describes the installation of the Reliant UNIX operating system.

Reliant UNIX 5.45

Reliant UNIX Operation

System Administrator Guide

Target Group

System administrators

Contents

Describes the operation of the Reliant UNIX operating system.

Reliant UNIX V5.44

System Administration within a Domain (Reliant UNIX)

System Administrator's Guide

Target Group

System administrators

Contents

The manual describes how to create and manage domains using *DomainAdmin*.

Reliant UNIX 5.45

Commands. User's Reference Manual

Description

Target Group

Users, system administrators

Contents

These manuals describe the user commands of the Reliant UNIX 5.45 operating system.

Reliant UNIX 5.44

System Administrator's Guide

System Administrator Guide

Target Group

System administrators

Contents

Introduces you to UNIX system administration. Among other information, the manual describes system administrator tools, UNIX system configuration and support, calculations of system usage as well as aspects of system security. The manual applies both for MIPS-based UNIX and for Intel-based UNIX.

Reliant UNIX 5.45

System Administrator's Reference Manual

Reference Manual

Target Group

System administrators

Contents

Describes commands and application programs for system maintenance, file formats and special files for system administration of the Reliant UNIX 5.45 operating system.

Reliant UNIX 5.44

System Administration and Hardware Configuration Using the SYSADM User Interface

System Administrator Guide

Target Group

System administrators

Contents

Describes operation of SYSADM, system administration with SYSADM (managing file systems and network services, installing software, system configuration, user administration), Config.

Reliant UNIX

OBSERVE V1.2

User Guide

Target Group

System administrators, programmers

Contents

Describes OBSERVE V1.2. The manual is written for system administrators responsible for setting up and supporting operation of OBSERVE, as well as for programmers who want to use the C interfaces offered with OBSERVE.

Reliant UNIX

SIDL M V1.2

Target Group

System administrators, service engineers

Contents

Describes the installation of the SINIX Distributed Lock Manager (SIDLM) and the management of the administration domains.