

English



BS2000/OSD

VM2000 V10.0

Virtual Machine System

User Guide

Edition September 2012

Comments... Suggestions... Corrections...

The User Documentation Department would like to know your opinion on this manual. Your feedback helps us to optimize our documentation to suit your individual needs.

Feel free to send us your comments by e-mail to:

manuals@ts.fujitsu.com

Certified documentation according to DIN EN ISO 9001:2008

To ensure a consistently high quality standard and user-friendliness, this documentation was created to meet the regulations of a quality management system which complies with the requirements of the standard DIN EN ISO 9001:2008.

cognitas. Gesellschaft für Technik-Dokumentation mbH

www.cognitas.de

Copyright and Trademarks

Copyright © Fujitsu Technology Solutions GmbH 2012.

All rights reserved.

Delivery subject to availability; right of technical modifications reserved.

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

EMC²®, Symmetrix®, CLARiiON CX, SRDF™, TimeFinder™, SnapView™, Engenuity™ and EMC ControlCenter™ are trademarks of EMC2 Corporation, Hopkinton/MA (USA).

The Xen[®] mark is a trademark of Citrix Systems, Inc., which manages the mark on behalf of the Xen open source community. The Xen[®] mark is registered with the U.S. Patent and Trademark Office, and may also be registered in other countries. Use of the Xen[®] mark on this product is not necessarily an indication of a relationship between or endorsement by Citrix Systems, Inc., the Xen community, or the Xen Advisory Board and the developers of this product, nor is it an indication that this product meets all the criteria for a “Faithful Implementation” of the Xen[®] hypervisor set out at <http://www.xen.org>.

Contents

1	Preface	13
1.1	Objectives and target groups of this manual	14
1.2	Summary of contents	15
1.3	Changes since the last edition of the manual	17
1.4	Notational conventions	18
1.5	Possible areas of application for VM2000	19
1.6	VM2000 characteristics	21
1.7	Linux or Windows guest systems (SQ servers)	22
2	Structure of VM2000	23
2.1	Terms of VM2000	23
2.2	Implementation of VM2000	27
2.2.1	Implementation on S servers	28
2.2.2	Implementation on SQ servers	30
2.3	Resource management under VM2000	35
2.4	Supported hardware and operating systems	36
2.5	Constraints	37
2.5.1	Function floppy disk or CD	37
2.5.2	Constraints that apply to the hardware	38
2.5.3	Constraints that apply to VM2000	39
2.5.4	Constraints that apply to guest systems	40
2.6	Differences between S and SQ servers	41

3	Installing VM2000	43
3.1	Generating and configuring the device peripherals	45
3.1.1	Device peripherals on S servers	45
3.1.2	BS2000 devices on the SQ server	47
3.2	Setting up the VM2000 monitor	48
3.3	Installing the VM2000 message files	51
3.4	Installing the VM2000 hypervisor files (S servers)	52
3.5	Setting the VM2000 parameters in the parameter service	53
3.6	Installing the configuration file for the monitor VM (S servers)	56
3.7	Preparing to use \$VMCONS	60
3.8	Installing the guest systems	61
3.9	Settings for VM2000 in X2000 (SQ servers)	62
3.9.1	Configuring VM2000	62
3.9.2	Creating KVP consoles to operate the guest system	63
3.9.3	Local and remote communication	64
4	Initializing and terminating VM2000	65
4.1	Initializing VM2000 operation	65
4.1.1	IPL for S servers	66
4.1.2	Initializing VM2000 operation (SQ servers)	68
4.2	Starting up the monitor system	69
4.2.1	Required components	69
4.2.2	Startup procedure in the monitor system	70
4.2.3	Example of startup in the monitor system	73
4.3	Terminating the monitor system and VM2000	79
4.3.1	Coordinated shutdown of VM2000	80
4.3.2	Shutdown in the monitor system	83
4.4	Loading, operating and terminating guest systems	84

5	Operating VM2000 and the virtual machines (VMs)	87
5.1	VM2000 Operating strategy	88
5.1.1	Operating levels of VM2000	88
5.1.2	Operating the virtual machines (ADMIN dialog from user task)	91
5.1.2.1	Privileges of the user task in the monitor system	91
5.1.2.2	Administration dialogs from privileged user tasks	92
5.1.2.3	Example of an ADMIN dialog from a user task	95
5.1.3	Operating the guest systems using BS2000 consoles	100
5.1.3.1	KVP consoles on S servers (via SCP 3970)	102
5.1.3.2	KVP consoles on SQ servers	107
5.1.4	VM2000 administration application \$VMCONS	110
5.1.4.1	Establishing a connection to \$VMCONS	112
5.1.4.2	Operating the virtual machines (ADMIN dialog via \$VMCONS)	115
5.1.4.3	Operating the guest systems (VC dialog via \$VMCONS)	117
5.1.4.4	Example of ADMIN and VC dialog via \$VMCONS	124
5.1.5	Automating VM2000 and VM administration	130
5.2	Operating a VM	136
5.2.1	Initializing a VM	138
5.2.2	Monitoring a VM with job variables (MONJV)	148
5.2.3	Managing devices and assignment sets	154
5.2.4	VM and VM group	154
5.2.5	Assigning a VM or a VM group to a CPU pool	154
5.2.6	Starting VM administration	155
5.2.7	Starting the guest system on a VM	156
5.2.8	Information services	159
5.2.9	Resource management	161
5.2.10	Suspending a VM/resuming operation	163
5.2.11	Moving a VM within main memory	166
5.2.12	Passing information to VM administrators	166
5.2.13	Modifying the VMCONS parameters	167
5.2.14	Terminating VM administration	167
5.2.15	Aborting the guest system on a VM	168
5.2.16	Shutdown in the guest system and of VM2000	170
5.2.17	Terminating a VM	174
5.3	Operation of VM groups (S servers)	176
5.3.1	Creating a VM group	178
5.3.2	Adding a VM to a VM group	179
5.3.3	Modifying the attributes of a VM in a VM group	180
5.3.4	Removing a VM from a VM group	180
5.3.5	Modifying the attributes of a VM group	180
5.3.6	Outputting information on a VM group	181

5.3.7	Deleting a VM group	181
5.3.8	Example of operating VM groups	182
5.4	Operating the whole VM2000 system	186
5.4.1	Types of command	187
5.4.2	Overview of commands for VM2000 operation	188
5.4.3	Privileged functions	190
5.4.4	Restrictions in VM2000 commands/functions	191
5.4.5	Extending and restricting the range of commands/functions	192
5.5	VM2000 in the guest systems	194
	ATTACH-DEVICE	
	Attach hardware units	197
	DETACH-DEVICE	
	Detach hardware units	198
	INCLUDE-DEVICE-CONNECTION	
	Add logical connections	199
	REMOVE-DEVICE-CONNECTION	
	Remove logical connections	200
	MODIFY-IO-UNIT	
	Modify the configuration properties of the input/output unit	201
	SHOW-SYSTEM-INFORMATION	
	Display information on the current system	202
	SHOW-DEVICE-CONFIGURATION	
	Display system configuration and availability	206
6	VM2000 management	211
6.1	CPU management	212
6.1.1	Managing and reconfiguring real and virtual CPUs	212
6.1.1.1	Real CPUs (processors)	213
6.1.1.2	Virtual CPUs	215
6.1.1.3	Reconfiguration of real and virtual CPUs	218
6.1.1.4	High availability with hot spare CPUs	220
6.1.1.5	Increasing capacity with extra CPUs	222
6.1.2	Managing CPU pools	225
6.1.2.1	Static CPU pools	225
6.1.2.2	Dynamic CPU pools	225
6.1.2.3	Number of attached real and virtual CPUs in the CPU pool	228
6.1.2.4	Creating a CPU pool	231
6.1.2.5	Adding CPUs to a CPU pool	232
6.1.2.6	Assigning a VM or VM group to a (different) CPU pool	232
6.1.2.7	Canceling the assignment of a VM or VM group to a CPU pool	232
6.1.2.8	Outputting information on a CPU pool	232

6.1.2.9	Deleting a CPU pool	233
6.1.2.10	Example of managing CPU pools	233
6.1.3	Planning distribution of the CPU capacity to the VMs	238
6.1.4	Scheduling procedures	249
6.1.4.1	Fixed CPU assignment (dedicated CPUs, S servers)	249
6.1.4.2	CPU assignment in the time-slicing procedure (S servers)	251
6.1.5	Example of CPU management	253
6.2	Reconfiguring Channels, controllers and paths (S server)	262
6.3	Memory management and reconfiguration	265
6.3.1	Managing main memory	265
6.3.1.1	Managing main memory on S servers	266
6.3.1.2	Managing main memory on SQ servers	269
6.3.2	Reconfiguring main memory	270
6.3.3	Global storage management (S server)	277
6.4	Peripheral device management	284
6.4.1	Assigning and removing devices of a VM	285
6.4.1.1	Explicit device assignment and removal	287
6.4.1.2	Assignment sets, implicit device assignment and release	288
6.4.1.3	Interplay between implicit and explicit device assignment	291
6.4.1.4	Usage mode of a device	292
6.4.1.5	Example of peripheral device management	294
6.4.2	Assigning and removing pubsets of a VM	305
6.4.2.1	Explicit pubset assignment and removal	307
6.4.2.2	Implicit pubset assignment and release	308
6.4.2.3	Interplay between implicit and explicit pubset assignment	310
6.4.2.4	Usage mode of pubsets	311
6.4.2.5	Notes on pubset specification in the VM2000 commands	313
6.4.2.6	Pubset reconfiguration	315
6.4.2.7	Mirror pubsets	318
6.4.2.8	Space provisioning (SPACEPRO) under VM2000	321
6.4.2.9	Example of pubset management	324
6.4.3	Special devices	336
6.4.4	Shared disk devices	338
6.4.5	Parallel access volume (PAV, S server)	345
6.4.6	Using the Highspeed Net Connect HNC (S server)	352
6.4.7	Notes on configuration for VM2000 operation	358
6.4.8	Use of IORM in VM2000 operation	359
6.4.9	Dynamic I/O configuration modification	361
6.4.9.1	Changing the configuration of channel devices via BS2000/OSD (S server)	361
6.4.9.2	Changing the configuration of devices via X2000	362
6.4.9.3	Changing the configuration of devices in the disk storage system	362

6.5	Backing up and restoring the VM2000 configuration	363
6.5.1	Backing up a VM2000 configuration	363
6.5.2	Modifying the backup of a VM2000 configuration	364
6.5.3	Restoring a VM2000 configuration	364
6.5.4	Backup file for a VM2000 configuration	367
6.5.4.1	Storage location	367
6.5.4.2	Structure of the contents	367
6.5.4.3	VM2000 components not backed up	371
6.5.4.4	Example of a backup file	372
6.6	VM2000 in the high-availability cluster (SQ servers)	376
6.6.1	VM2000 information on the VMs in the HA cluster	377
6.6.2	General conditions for Live Migration under VM2000	378
6.6.3	VM2000 information during Live Migration of a VM	379
6.7	Password protection	380
6.8	Accounting in VM2000	381
6.8.1	Accounting record for devices	384
6.8.2	Accounting record for a VM	386
6.9	Hardware error logging under VM2000	388
6.10	Monitoring under VM2000	391
6.10.1	Measurement with /SHOW-VM-STATUS	391
6.10.2	Measurement with the openSM2 measurement monitor	392
7	Error handling in VM2000	393
7.1	Types of error in VM2000	393
7.1.1	Errors in the carrier system (SQ servers)	394
7.1.2	VM2000 errors	395
7.1.3	Guest system errors	398
7.2	Restart routines in VM2000	401
7.2.1	Automatic restart following termination of the VM2000 hypervisor (S servers)	401
7.2.2	Automatic restart following termination of the VM2000 Agent	402
7.2.3	Automatic restart after SETS in the monitor system	402
7.2.4	Restarting the monitor system via SVP	403
7.2.5	Restarting a task after an error in a system task of VM2000	405
7.3	VMDUMP utility routine	406
7.4	Other diagnostic aids	409

7.5	Diagnostic file SLEDFILE	410
7.5.1	Creating SLEDFILES during VM2000 operation	410
7.5.2	Analyzing SLEDFILES with DAMP	412
8	VM2000 commands	415
8.1	Overview of commands for VM2000 operation	416
8.1.1	Table of VM2000 commands	417
8.1.2	Grouping of VM2000 commands according to function	419
8.2	Command return codes	423
8.3	Guaranteed messages	424
8.4	SHOW output in S variables	425
8.5	Description of the commands	426
	ADD-VM-DEVICES	
	Allocate devices to a VM	428
	ADD-VM-RESOURCES	
	Assign global storage to VM (S servers)	433
	ADD-VM-TO-VM-GROUP	
	Add a VM to a VM group (S servers)	436
	ASSIGN-VM-GROUP-TO-CPU-POOL	
	Assign a VM group to a CPU pool (S servers)	438
	ASSIGN-VM-TO-CPU-POOL	
	Assign a VM to a CPU pool	440
	ATTACH-VM-RESOURCES	
	Attach real CPUs	442
	BEGIN-VM-DIALOG	
	Start administration or operation	446
	CALL-VM-PROCEDURE	
	Start a VM2000 procedure file	449
	CREATE-VM	
	Initialize a VM	452
	CREATE-VM-ASSIGNMENT-SET	
	Create assignment set	463
	CREATE-VM-CPU-POOL	
	Create a CPU pool	464
	CREATE-VM-GROUP	
	Create VM group (S servers)	465
	DELETE-VM	
	Terminate a VM	467

DELETE-VM-ASSIGNMENT-SET	
Deletes an assignment set	469
DELETE-VM-CPU-POOL	
Delete CPU pool	470
DELETE-VM-GROUP	
Delete VM group (S servers)	471
DETACH-VM-RESOURCES	
Detach real CPUs	472
END-VM-DIALOG	
Terminate administration or operation	476
EXTEND-VM-MEMORY	
Increase main memory for a VM	478
HOLD-VM	
Suspend VM	482
MODIFY-VM-ATTRIBUTES	
Modify attributes of a VM	484
MODIFY-VM-DEVICE-ATTRIBUTES	
Modify device attributes	491
MODIFY-VM-DEVICE-USAGE	
Modify usage mode of the devices of a VM	495
MODIFY-VM-GROUP-ATTRIBUTES	
Modify attributes of a VM group (S servers)	499
MODIFY-VMCONS-PARAMETERS	
Modify parameters of \$VMCONS	501
MOVE-VM	
Move a VM within main memory (S servers)	503
REDUCE-VM-MEMORY	
Reduce main memory for a VM	506
REMOVE-VM-DEVICES	
Deallocate devices from VM	509
REMOVE-VM-FROM-VM-GROUP	
Remove VM from VM group (S servers)	514
REMOVE-VM-RESOURCES	
Cancel assignment of global storage to VM (S servers)	516
RESET-VM	
Aborts the guest system on a VM	519
RESTORE-VM-CONFIGURATION	
Restore the VM2000 configuration	521
RESUME-VM	
Resume VM operation	523
SAVE-VM-CONFIGURATION	
Save the VM2000 configuration	525
SEND-VM-MESSAGE	
Pass messages to VM administrators	527

SHOW-VM-ASSIGNMENT-SET	
Output assignment sets	529
SHOW-VM-ATTRIBUTES	
Output attributes of a VM	532
SHOW-VM-CPU-POOL	
Output attributes of a CPU pool	549
SHOW-VM-DEVICE-STATUS	
Output detailed device information	556
SHOW-VM-GROUP	
Output attributes of VM group (S servers)	573
SHOW-VM-RESOURCES	
Output status of VM2000 resources	579
SHOW-VM-STATUS	
Output VM2000 monitored data	607
SHUTDOWN-VM	
Execute a shutdown in the guest system and for VM2000	623
START-VM	
Start a guest system on a VM	627
SWITCH-VM-CPU	
Delete CPUs from a CPU pool and add them to another CPU pool	631
SWITCH-VM-DEVICES	
Deallocate devices from one VM, allocate them to another	635
Glossary	641
Related publications	647
Index	651

1 Preface

In order to meet the many and diverse demands placed on information technology, it is becoming necessary to operate different system environments simultaneously on the BS2000/OSD servers. In former times this problem has been solved by installing multiple servers or by serializing operation and making use of night-time hours.

The **VM2000 virtual machine system** permits the simultaneous operation of different, completely self-contained system environments on a **single BS2000/OSD server** with almost the same performance as in native mode (operation without VM2000). VM2000 enhances system versatility and improves the capacity utilization of the server.

The systems on which IT processes are executed must adhere to the highest standards of security, reliability, scalability and performance. VM2000 has the system attributes needed for this in abundance and offers functions that provide the user with a variety of system deployment scenarios for improving availability. This applies both to individual servers and clusters of servers.

VM2000 V10.0 supports all current S and SQ servers.

1.1 Objectives and target groups of this manual

This manual is aimed at all those who use the VM2000 virtual machine system on their BS2000/OSD server or have to make preparations for its use.

It is aimed specifically at:

- the VM2000 administrator, who operates the whole VM2000 system
- the VM administrator, who operates a virtual machine
- systems support and operating

Familiarity with the operation and management of the BS2000/OSD operating system and knowledge of the operation of the BS2000/OSD server are essential for an understanding of this manual.

The central manual for operating SQ servers is the manual “SQ Series Business Servers, Operation and Administration” [18]. In addition to the concepts and functions for SQ servers, this manual also describes in detail how to work with the SQ manager, the user interface of the SQ servers.

1.2 Summary of contents

VM2000 V10.0 is the evolutionary development and combination of the software products VM2000 V9.0 (S servers) and VM2000 V9.5 (SQ servers)

When different hardware-related approaches to a topic exist, each approach is described in a separate section.

The function and command descriptions contain information which only applies for particular server series. This information is labeled, see [section “Notational conventions” on page 18](#).



A comparative overview of the differences between the various server series is provided in the [section “Differences between S and SQ servers” on page 41](#).

Main chapters of the manual and their target group

The [chapter “Structure of VM2000”](#) explains the key terms used in the manual and provides a basic overview of VM2000. It is aimed at all users of VM2000.

The [chapter “Installing VM2000”](#) describes the components of VM2000 and the preparations that have to be carried out in order to use VM2000. It is aimed at system administrators.

The [chapter “Initializing and terminating VM2000”](#) describes how to initialize and terminate VM2000 and guest systems under VM2000. It is aimed at VM2000 administrators and system administrators.

The [chapter “Operating VM2000 and the virtual machines \(VMs\)”](#) describes the operating strategy for VM2000 and the functions of the VM2000 commands. It is aimed at VM2000 administrators and VM administrators.

The [chapter “VM2000 management”](#) describes the administration tasks involved with VM2000, such as resource management, accounting and password protection. It is aimed at VM2000 administrators and system administrators.

The [chapter “Error handling in VM2000”](#) describes error classes that can occur in operation with VM2000 and the diagnostic tools that are available. It is aimed at VM2000 administrators and system administrators.

The [chapter “VM2000 commands”](#) contains the syntax and semantics of all of the VM2000 commands. It is aimed at VM2000 administrators and VM administrators.

Readme file

The functional changes to the current product version and revisions to this manual are described in the product-specific Readme file.

Readme files are available to you online in addition to the product manuals under the various products at <http://manuals.ts.fujitsu.com>. You will also find the Readme files on the Softbook DVD.

Information under BS2000/OSD

When a Readme file exists for a product version, you will find the following file on the BS2000 system:

```
SYSRME.<product>.<version>.<lang>
```

This file contains brief information on the Readme file in English or German (<lang>=E/D). You can view this information on screen using the `/SHOW-FILE` command or an editor. The `/SHOW-INSTALLATION-PATH INSTALLATION-UNIT=<product>` command shows the user ID under which the product's files are stored.

Additional product information

Current information, version and hardware dependencies, and instructions for installing and using a product version are contained in the associated Release Notice. These Release Notices are available online at <http://manuals.ts.fujitsu.com>.

1.3 Changes since the last edition of the manual

The manual for VM2000 V10.0 describes the use of VM2000 on S and SQ servers. It is based on the VM2000 V9.0A and VM2000 V9.5A manuals.

Compared to the previous version, the manual contains the following major changes which were introduced with VM2000 V10.0:

- SX servers are no longer supported in BS2000/OSD V9.0 and higher. Information on these has been removed from this manual.
- VMs on SQ servers can be configured with up to 32 virtual CPUs.
- CPU pools are also supported for SQ servers.
- New [section “Backing up and restoring the VM2000 configuration”](#).
- New commands `/SAVE-VM-CONFIGURATION` and `/RESTORE-VM-CONFIGURATION`.
- New [section “VM2000 in the high-availability cluster \(SQ servers\)”](#).
- New operand value `INFORMATION=*HA-CLUSTER` in the `/SHOW-VM-DEVICE-STATUS` and `/SHOW-VM-RESOURCES` commands.
- New VM state `MIGRATING` (`/SHOW-VM-ATTRIBUTES` and `-RESOURCES`).

Compared to the previous version, the manual contains the following major changes which were already introduced with VM2000 V9.5:

- The VM2-MON subsystem has a separate parameter file. This also contains the parameters of `$VMCONS`.
- In the [chapter “Initializing and terminating VM2000”](#) first the initialization of VM2000 operation is described in place of "system initialization" and then the startup of the monitor system.
- The term “operator terminal” is no longer used. The term “BS2000 console” is used whenever the operation of the monitor system or of another guest system is meant.
- The term “domain” (synonym for the main memory (area) of a VM) is no longer used. In Xen the term is used as a synonym for "virtual machine" (e.g. domain 0).
- Operation of the S Connection Director has been omitted since BS2000/OSD V8.0.
- The following commands have new operands and operand values:
 - `/BEGIN-VM-DIALOG`, new operand value `TYPE=*STD`
 - `/CREATE-VM`, new operand `MAX-MEMORY-SIZE`
 - `/CREATE-VM`, operand `PROCESSOR`, new operand value `*EXTRA-AND-NORMAL`

- The "**Messages**" chapter is no longer updated. You will find the messages using an HTML application on our manual server (URL: <http://manuals.ts.fujitsu.com>) instead of in the previous manual "System Messages" and on the DVD "BS2000/OSD SoftBooks".

1.4 Notational conventions

The following abbreviations are used in this manual:

- **S server** (e.g. S210) for the business servers of the S series (/390 architecture)
- **SQ server** (e.g. SQ200) for the business servers of the SQ series (X86-64 architecture)

The strings `<date>`, `<time>` and `<ver>` in the examples stand for the current outputs for date, time and version when the examples are otherwise independent of date, time and version.



Because of the fact that the product name comes up so frequently in this manual, the current version of VM2000 is often shortened to VM2000.

The following notational conventions are used in this manual:



This symbol indicates an important passage

- ▶ The input symbol identifies activities to be executed (e.g. entries via the keyboard).

You will find a description of the SDF syntax in the "Commands" manual [12].

Where references are made to other publications, the titles are abbreviated. The complete title of each publication referred to by a number in square brackets is given under "Related publications".

In the examples, `bo1d` type indicates a user entry and this `typewriter` font is used for system outputs.

The command descriptions contain a great deal of information which only applies for particular server series. This information is labeled as follows:



This comment applies only for S servers



This comment applies only for SQ servers

1.5 Possible areas of application for VM2000

Separate systems for production and backup

Data centers that offer computing power for production or backup without VM2000 have to have several different servers for different system environments.

With VM2000 it is possible to offer the same service on a single (or a few) high-performance server. This makes more sophisticated capacity planning possible and allows savings to be made in terms of floor space and personnel.

The same benefits can also be gained by customers who themselves require different production environments. Sensitive applications can be isolated in a secure environment with VM2000.

Parallel operation of production, development and testing

A separate system environment is sometimes necessary for the development and testing of user systems for the following reasons:

- System functions not used to date are addressed
(the application may have been programmed incorrectly or run inefficiently)
- System modifications are being tested
(applications which have not been fully tested can disturb productive operation, e.g. through high resource utilization)
- Errors are caused intentionally or unintentionally in test cases

VM2000 permits what are known as “virtual block times”, i.e. selected systems can be run on a server but do not affect the other systems running at the same time, even in the event of system errors. In conjunction with resource allocation to the individual systems, it is also possible to run multiple test environments in parallel with production.

Version changes

In order to prepare themselves for changes to the versions of the operating system or system-related software, personnel must get to know how to install and use the new versions. It must be ensured that production runs smoothly in the new environment.

VM2000 supports you in:

- reducing night-time testing and weekend working when it comes to operating the new version and testing for production
- converting production to the new system environment step by step, in as far as this is possible with separate, independent bodies of data

VM2000 in the data center

In addition to the savings potential in terms of operating staff and computer space requirements, VM2000 also offers a high degree of cost-effectiveness and optimum use of resources through the virtualization of the CPU, main memory and global storage.

VM2000 supports accounting for the computer performance and resources provided for a customer:

- **Accounting according to usage**
The VM-specific accounting records provided by VM2000 show how many resources were used for how long.
- **Definition of Service Level Agreements**
The customer is guaranteed a certain CPU capacity with a price based on RPF. The CPU capacity of the VM is restricted to the required value using the “maximum CPU utilization” attribute (`MAX-CPU-UTILIZATION` operand).
- **Creating CPU pools and dedicated CPUs**
CPUs can be grouped in CPU pools. Only the VMs which are assigned to a CPU pool utilize the CPUs of this pool. The CPUs and VMs of the pool form a subsystem of the server, and this is made available to a customer.

If the CPU pool contains enough real CPUs, the scheduling procedure “fixed CPU assignment (dedicated CPUs)” reaches the highest level of performance (S servers).
- **Creation of VM groups (S servers)**
For customers with multiple VMs it is possible to create a VM group and define an arrangement based on Service Level Agreements. In the first step the CPU capacity and CPU utilization for the VM group are defined. In the second step the performance distribution within the VM group is defined. When distributing the CPU capacity which a VM in a VM group does not use, the scheduling procedure gives preference to the other VMs in the VM group.
- **Granularity setting of CPU-QUOTA and MAX-CPU-UTILIZATION**
The two attributes for controlling the VM's performance can be specified with two decimal places. This means that very high server definitions are possible in the one-digit RPF area up to one percent of the CPU capacity.

VM2000 in the availability cluster

The aim of an availability cluster of BS2000/OSD servers under BS2000/OSD is to continue productive operation on a redundant server if a server or application should crash.

VM2000 in the HIPLEX cluster ensures:

- **system availability**

If the guest system on which the application is running crashes, the entire system environment with the application can be made available on another server in the shortest possible time.

- **application availability**

If the application crashes, it can continue to run on a backup guest system on the same server or be switched to another server in the shortest possible time.

In regular operation the backup guest system is actively redundant and occupies few resources.

In the event of an error the backup guest system's main memory is extended through dynamic main memory reconfiguration in VM2000, and HIPLEX moves the productive application to the backup guest system.

1.6 VM2000 characteristics

VM2000 is characterized by the following features:

- **Concurrent** execution of up to 15 guest systems on a **single** BS2000/OSD server. Guest systems can be the operating systems released for this (see [page 36](#)).
The guest systems are completely independent of one another.
 - it is not possible to access memory areas of other guest systems.
 - errors within one guest system - even if they result in a system crash - are of no relevance for the other guest systems.
- Scope of functions identical to that of BS2000 native operation.
 - The functionality of guest systems running under VM2000 corresponds to BS2000 native operation as far as the command set, the data communication capability in networks, and the debugging and diagnostic aids are concerned.
 - For accounting purposes, the job accounting data of every guest system can be centrally collated.

- Performance almost the same as BS2000 operation without VM2000.

The performance of the guest systems under VM2000 is similar to that of native operation. The excellent performance values are a result of the following measures:

- The guest systems execute directly on the host, i.e. only one minor emulation is required.
- Memory areas are permanently assigned to a given guest system (thus eliminating the need for additional address translation).
- The devices are assigned to the guest system, i.e. no additional loss in performance results from device management under VM2000.

Further information on the use of VM2000 in terms of performance is provided in the “Performance Handbook” [17].

1.7 Linux or Windows guest systems (SQ servers)

On SQ servers not only VMs with a BS2000 guest system can be configured, but also VMs with a Linux or Windows guest system, the so-called XenVMs.

XenVMs are managed exclusively via the SQ manager, see the manual “SQ Series Business Servers, Operation and Administration” [18].

They are not known in VM2000. Only when global resources are displayed are they included under the keyword `FOREIGN`.

2 Structure of VM2000

2.1 Terms of VM2000

VM2000, virtual machine, guest system

The VM2000 virtual machine system is a software product which offers **several virtual machines (VMs)** on a **single** BS2000/OSD server.

From the user's point of view, a virtual machine is a BS2000/OSD server on which his/her **guest system** can be operated.

Monitor VM, monitor system

A special BS2000 guest system, the **monitor system**, in which the **VM2000 monitor**, the administration application **\$VMCONS** and the other VM2000 components are embedded, is provided to execute and manage VM2000. The monitor system runs on a special VM, namely the **monitor VM**.

The monitor VM not only serves to control the operation of virtual machines but, like any other VM, can also be used for the standard applications described on [page 19](#).

VM2000 monitor

The VM2000 monitor is the central part of VM2000 for operating the virtual machines using VM2000 commands. It implements communication both between the user and VM2000, and between VM2000 and the hardware-related components, in particular the hypervisor (see the next page).

The VM2000 monitor is implemented in the monitor system as the VM2-MON subsystem.

VM2000 administration application \$VMCONS

Operation of the VMs is normally conducted from a privileged user task in the monitor system. BS2000 consoles for the guest system are as a rule available to operate the guest systems. See [section “VM2000 Operating strategy” on page 88](#).

The administration application \$VMCONS (optional, see [page 110](#)) can be used as an alternative to these recommended administration and operating media.

VM2000 dump

A special VM2000 dump for diagnostics can be created using the VMDUMP utility routine.

In the event of an error, the VM2000 dump is created automatically by VM2000 using the VMDUMP utility routine.

The VM2000 dump is not required for regular operation of VM2000.

VM2000 in the BS2000 guest systems

Each BS2000 guest system has communication interfaces to VM2000 in order to receive information on VM2000 operation and to introduce VM2000-specific actions in the guest system (implicit device assignment, for example, see [page 288](#)).

These interfaces are described in detail in [section “VM2000 in the guest systems” on page 194](#).

Hypervisor

A hypervisor controls the execution of the guest systems on the VMs. In particular it virtualizes the global resources CPU and main memory and starts execution of the operational guest system’s CPUs on the real CPUs (scheduling).

- S On S servers the VM2000 hypervisor is a separate load module of VM2000 which is loaded (automatically) when VM2000 operation is initialized.
- SQ On SQ servers the Xen hypervisor performs this role. Some of the hypervisor tasks are performed by the carrier system X2000.

Hardware-software interface (HSI)

In technical terms, when operating virtual machines the hardware-software interface (HSI) which the BS2000/OSD server offers for an operating system is copied and made available for all VMs and their guest systems (see [figure 1 on page 26](#)). The (copied) HSI for a VM or a guest system is referred to as a **virtual HSI**.

Differences between S and SQ servers with respect to the HSI

S S servers are systems with a /390 architecture.

The BS2000/OSD operating system uses the HSI of the S server (**the real HSI**) directly here.

SQ SQ servers are systems with an X86-64 architecture.

The BS2000/OSD operating system (in the OSD/XC software package) uses the nonprivileged, real HSI of the SQ server here and the privileged HSI which is provided by the firmware.

For further information on the technical structure of the SQ servers and on the system environment, see the manual “SQ series Business Servers, Concepts and Functions” [18].

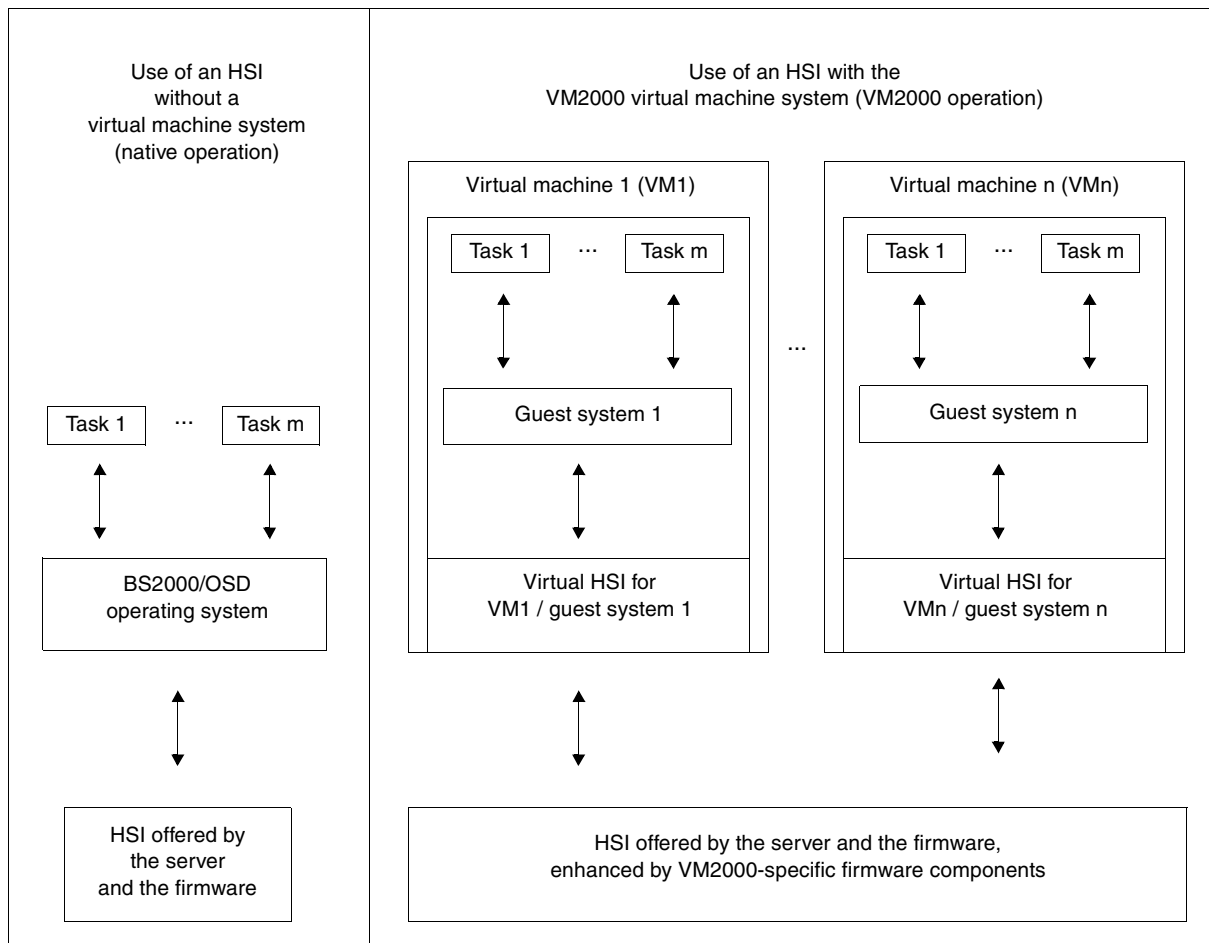


Figure 1: Use of the HSI of a BS2000/OSD server without and with VM2000

2.2 Implementation of VM2000

The technical implementation of VM2000 depends on the server series. It is described in short in the sections below.

There are two fundamentally different ways of implementing VM2000:

S On S servers VM2000 controls the hardware of the server.

The VM2000 monitor manages all the VMs and provides its functions via the VM2000 user interface.

A VM2000 hypervisor controls the execution of all guest systems on the VMs. Sophisticated scheduling mechanisms ensure that the guest systems run as efficiently as possible.

SQ On SQ servers the X2000 carrier system controls the hardware of the server.

The VM2000 monitor manages the VMs with the guest system BS2000/OSD (**BS2000 VM**) and provides its functions via the VM2000 user interface. Selected functions (e.g. IPL or shutdown) for BS2000 VMs can also be executed via the SQ manager, the user interface of the SQ servers. In addition, the SQ manager can be used to configure further VMs in which Linux or Windows systems execute as guest systems (**XenVMs**). These are managed only by the SQ manager, not by VM2000.

The Xen hypervisor virtualizes the global resources CPU and main memory, controls the execution of all VMs (scheduling), and performs load balancing for CPU utilization.

The central manual for operating SQ servers is the manual “SQ Series Business Servers, Operation and Administration” [18]. A detailed description of working with the SQ manager is also provided there.

2.2.1 Implementation on S servers

For VM2000 operation, VM2000 functions have been added to the HSI offered by the firmware of the S servers. The extended HSI is referred to as the **VM2000 HSI** (see [figure 2](#)).

The VM2000 hypervisor runs on the **VM2000 HSI**. It is the part of VM2000 which controls the execution of guest systems on the virtual machines (the scheduling).

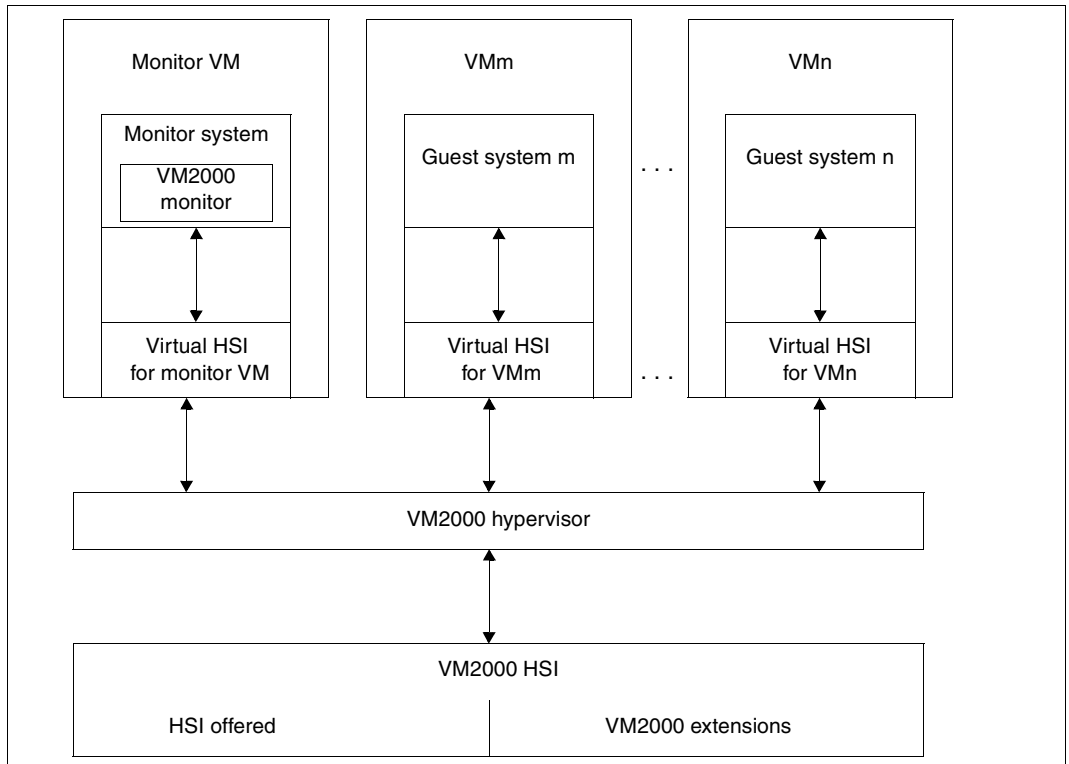


Figure 2: Structure of VM2000 on S servers

The functions of the VM2000 hypervisor on S servers are:

- allocate the CPUs to the individual VMs
- handle global interrupts
- control machine error recovery
- emulate certain privileged instructions
- coordinate I/O operations for all shared devices
- process monitor jobs

In the VM2000 HSI, two processor states are possible (see [figure 3](#)):

- **VM mode.**
The guest systems run in this mode.
- **hypervisor mode.**
The VM2000 hypervisor runs in this mode. Additional privileged commands only permitted in this mode are also available.

Each mode has a **context**, which is loaded into the servers hardware registers each time the processor state is activated. This context consists of

- the registers for the processor states of the mainframe
- VM-specific registers and information.

In the context, a **VM index** which identifies the VM is entered for every VM. The monitor VM contains always the VM index 1.

The transition from VM mode to VM2000 hypervisor mode is initiated by **global** interrupts, e.g. expiration of the time slice for the VM. In the case of global interrupts the context of the VM is stored and the VM2000 hypervisor context is loaded, thereby activating the VM2000 hypervisor.

There are also **local** interrupts, which do **not** activate the VM2000 hypervisor but are processed within the VM.

You can switch from VM2000 hypervisor mode to VM mode by issuing a privileged VM2000 hypervisor switch instruction.

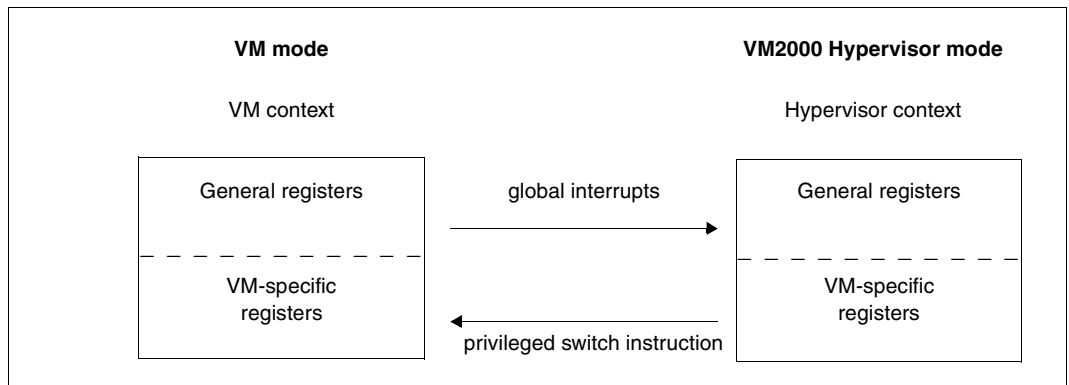


Figure 3: Processor states and context on S servers

2.2.2 Implementation on SQ servers

Domains

On SQ servers one or more operating systems can be started in virtual machines, the so-called **domains**.

The first domain which is started automatically is of particular importance. This domain, which is called **domain 0** (“Dom0”), is privileged and is used for management purposes. The carrier system X2000 executes in domain 0.

The **Xen hypervisor** and the IO drivers in domain 0 operate and virtualize the server unit. The Xen hypervisor controls execution of all the domains (scheduling).

Different types of operating systems, e.g. BS2000/OSD or Linux and Windows systems, can run in the nonprivileged domains (“DomU”).

Carrier system X2000

The X2000 carrier system executes under the Linux operating system in domain 0.

X2000 contains functions for supporting BS2000/OSD. It provides the operating functionality for devices (the /390 device emulation), and executes the physical inputs/outputs.

X2000 also implements the interface for operating the VMs via the SQ manager or the Customer Support.

VM2000

BS2000 VMs can be configured and operated only when the software product VM2000 is installed and configured on the SQ server.

Before VM2000 operation is initiated on SQ servers, the “VM2000” operating mode must be set using the SQ manager. In addition to domain 0, another special domain with the domain name `MONITOR`, the monitor VM, is then configured. The monitor system, which is a BS2000 guest system in which the VM2000 monitor and the other software components of VM2000 are embedded, executes on the monitor VM.



The terms domain and virtual machine can be regarded as synonyms for VM2000 operation. In this manual a distinction is consequently only made in important places. As a rule the term virtual machine (VM) is used. When no more detailed explanation is provided, a VMs with a BS2000 guest system (BS2000 VM) is meant.

The VM2000 functions and commands which are offered on SQ servers for BS2000 VMs are described in this manual. The VM2000 monitor enables further BS2000 VMs to be configured and operated.

The BS2000 guest systems can work together with VM2000, see [section “VM2000 in the guest systems” on page 194](#).

The VM2000 monitor operates the BS2000 VMs, but not the XenVMs. XenVMs are only operated via the SQ manager, see the manual “SQ Series Business Servers, Operation and Administration” [18].

For hardware-related implementation of the functions, VM2000 utilizes the carrier system X2000 in domain 0. For communication purposes X2000 realizes another software component, the VM2000 Agent.

VM2000 Agent

The VM2000 Agent executes in domain 0. It is started only in VM2000 operation.

On the one hand the VM2000 Agent converts the requests of the VM2000 monitor and of the guest systems (the hypervisor calls) to corresponding administration or guest system requests. On the other hand the VM2000 Agent also registers events which are important for VM2000 operation and informs the VM2000 monitor and the BS2000 guest systems affected of these.

Firmware component of a VM

To execute in an environment with X86-64 architecture, each BS2000 guest system requires a firmware component which converts the /390 code to the relevant X86-64 code (CISC-FW) and offers the privileged HSI for the BS2000 guest system (HAL).

On SQ servers each VM has such a firmware component.

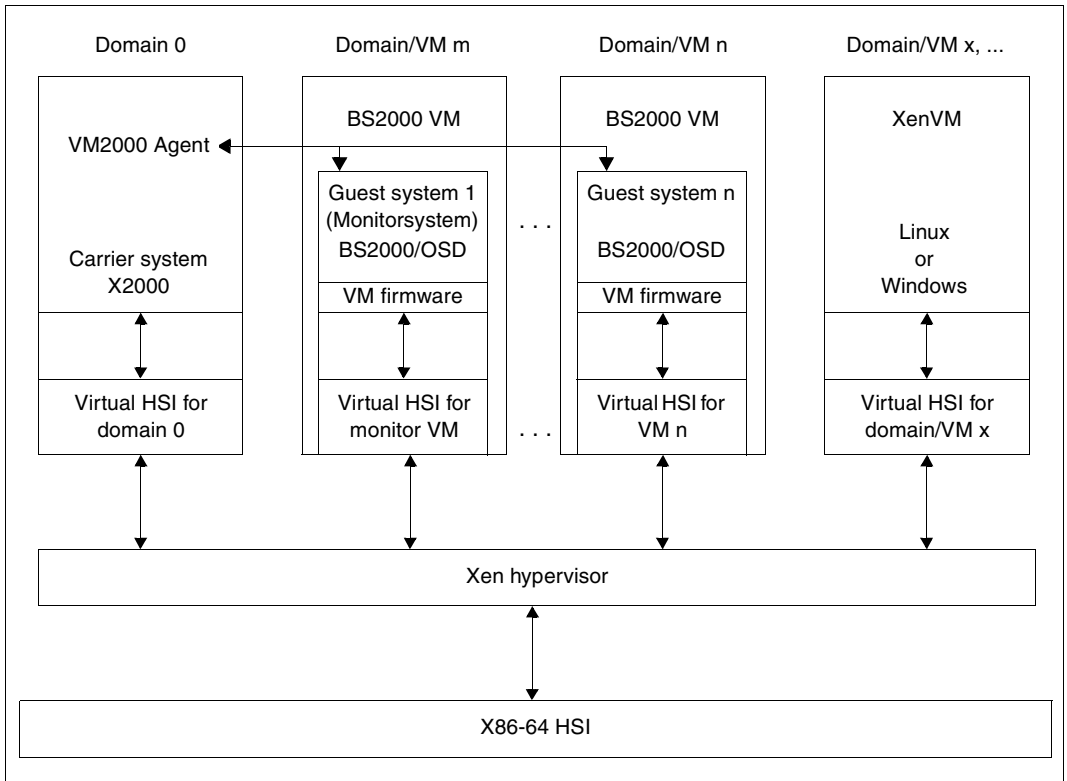


Figure 4: VM2000 on SQ servers

Operating levels

Actions for the BS2000 VMs can be initialized on SQ servers from various operating levels:

- The full VM2000 functionality is available to the VM2000 and VM administrators via the user interface of VM2000 (see [page 88](#)). The VM2000 commands operate and manage the BS2000 VMs, but not the XenVMs.
- The “start guest system”, “shut down guest system”, “abort VM” and “migrate VM” functions as well as the information outputs are available to the administrators of the SQ server via the SQ manager (the user interface of the SQ servers), also for BS2000 VMs.

Information on global resources in VM2000 commands

The VM2000 commands also supply information relating to resources which the BS2000 VMs share with the XenVMs (main memory, CPU performance):

- `/SHOW-VM-CPU-POOL` supplies information on all CPU pools of the Server Unit. This also includes information on the CPU pools of domain 0 and for the XenVMs
- The output of `/SHOW-VM-RESOURCES INFORMATION=*MEMORY,ATTRIBUTE=*ALL/*ASSIGNED` also supplies the amount of main memory occupied by the XenVMs and the domain 0 (FOREIGN VMS).
- `/SHOW-VM-RESOURCES INFORMATION=*HA-CLUSTER` also supplies the size of the main memory occupied by the XenVMs (FOREIGN VMS) (but not that of domain 0).
- `/SHOW-VM-RESOURCES INFORMATION=*CPU/*CONFIGURATION` supplies information on all CPUs of the Server Unit, their status and the assignment to CPU pools. This also includes information on the CPU pools of domain 0 and for the XenVMs
- `/SHOW-VM-STATUS INFORMATION=*TIMER` also supplies information on the load on domain 0 in `*POOL0`.

Main memory

The SQ server's main memory contains a fixed portion for domain 0. The remaining main memory can be used for BS2000 VMs or XenVMs.

The planning for the assignment of the real main memory to domain 0, for VM2000 operation and for the XenVMs is the task of the SQ server administration.

Information on the assignment of the main memory is provided by `/SHOW-VM-RESOURCES INFORMATION=*MEMORY`.

A BS2000 VM obtains the main memory in a granularity of 2 Mbyte or more. The main memory of a VM is not contiguous in the server's main memory (as it is on S servers), but scattered.

A small part of the main memory of a BS2000 VM is required for the firmware component of the VM. The VM's firmware component continues to emulate a contiguous main memory starting with address 0 for a BS2000 guest system.

See also [section "Managing main memory on SQ servers" on page 269](#).

CPUs, CPU pools

The real CPUs of the SQ server are subdivided into static CPU pools bei der Systemeinleitung des SQ-Servers. This subdivision is achieved by means of licenses. It can also be viewed using the VM2000 command `/SHOW-VM-RESOURCES INFORMATION=*CPU/*CONFIGURATION`.

In addition, further, unused real CPUs can exist (depending on the hardware and licenses) which are not assigned to a CPU pool. These unused CPUs automatically become cold spare CPUs.

The following static CPU pools are defined for VM2000 operation (the equivalent names in the VM2000 outputs and in the manual “SQ series Business Servers, Concepts and Functions” [18] are specified in parentheses):

- Pool of domain 0 (*POOL0, Pool-0).
It contains the CPUs for domain 0, i.e. for the X2000 carrier system.
- Standard CPU pool (*STDP00L, bs2_pool).
It contains the CPUs for VM2000 operation, possibly enhanced with extra CPUs.

The number of CPUs in the standard CPU pool is the same as the number of BS2000 CPUs of the SQ server model, including the extra CPUs (license required). VM2000 manages and reconfigures only the CPUs of the standard CPU pool, see [page 213](#).
- Foreign pool and Application pool (*FOREIGN, lw_pool + ap_pool).
It contains the CPUs for the XenVMs and for special applications (license required).

With VM2000 further (dynamic) CPU pools can be created. See [section “Managing CPU pools” on page 225](#).

Devices

The real devices in the periphery are not directly visible for the monitor system and the other BS2000 guest systems. BS2000 guest systems know only the **BS2000 devices** emulated in X2000. See [section “BS2000 devices on the SQ server” on page 47](#).

2.3 Resource management under VM2000

VM2000 manages the following resources:

CPU management

These management tasks comprise the following

- Management and reconfiguration of the real CPUs which are available for VM2000 operation (see [page 213](#)) and of the VMs' virtual CPUs (see [page 215](#))
- Management of the CPU pools, see [page 225](#)
- Planning and controlling the distribution of CPU capacity to the VMs, see [page 238](#)
- Assigning CPUs to a VM (scheduling, S servers), see [page 249](#)

Reconfiguration of channels, control units and paths

See [page 262](#) (S server).

Management and reconfiguration of the storage media

Memory management under VM2000 involves

- [management of the main memory](#), see [page 265](#).
- [reconfiguration of the main memory](#), see [page 270](#).
- [management of global storage \(S servers\)](#), see [page 277](#).

Peripheral device management

The following are given particular consideration in peripheral device management:

- [Assigning and removing devices of a VM](#), see [page 285](#).
- [Assigning and removing pubsets of a VM](#) (devices of the pubset), see [page 305](#)
- [Special devices](#), see [page 336](#).
- [Shared disk devices](#), see [page 338](#).
- [Parallel access volume \(PAV, S server\)](#), see [page 345](#).
- [Using the Highspeed Net Connect HNC \(S server\)](#), see [page 352](#).
- [Dynamic I/O configuration modification](#), see [page 361](#).

2.4 Supported hardware and operating systems

VM2000 V10.0 can be used on the current S and SQ servers with the carrier system required for this purpose.



Further BS2000/OSD servers will be released separately at different dates. For further information on how to use VM2000 on these models, see the product announcements from Fujitsu.

VM2000 V10.0 operates all peripheral devices that are supported as of BS2000/OSD V6.0 (S servers) respectively OSD/XC V4.0 (SQ servers).

VM2000 V10.0 is executable in a monitor system with BS2000/OSD V8.0 (for SQ servers in the software package OSD/XC V4.0) or BS2000/OSD V9.0 (OSD/XC V9.0).

Migration from VM2000 V9.0 or V9.5 to VM2000 V10.0 is possible on a BS2000/OSD-BC V8.0 monitor system without changing the monitor system.

The following guest system software can run under VM2000 V10.0:

- BS2000/OSD V6.0 (S servers)
- BS2000/OSD V7.0 (S servers)
- BS2000/OSD V8.0 (for SQ servers in the OSD/XC software package as of V4.0)
- BS2000/OSD V9.0 (for SQ servers in the OSD/XC software package as of V9.0)
- Successor versions to BS2000/OSD-BC V9.0, provided they have been released for use under VM2000 V10.0 (see release notices)
- self-loading systems (e.g. SLED) of the above versions of the operating system

These guest systems can be operated in the usual way under VM2000. There are various means available for operating the guest systems under VM2000 (see [page 89](#)).



You need a special release to use older versions of BS2000/OSD as the guest system on S servers.

2.5 Constraints

2.5.1 Function floppy disk or CD

Special firmware is required to operate the guest systems via KVP consoles on the SKP 3970-20 of the S servers. This firmware is provided by means of special function floppy disks or CDs.

- On S servers with SKP 3970-40 or -50 the function floppy disks and CDs are no longer required.
- Such function floppy disks or CDs are not required on SQ servers.

S server with SKP 3970-20

The function floppy disk `VM_OP` is a floppy disk containing firmware for operating VM2000 via SCP. It permits the local and remote KVP consoles connected to the SCP to be used for operating the monitor system and the guest systems, see [page 102](#). Both permanent assignment of the KVP console to one guest system each and a central KVP console for multiple guest systems are possible here.

2.5.2 Constraints that apply to the hardware

BS2000/OSD servers exhibit certain differences from the full functional scope of VM2000 due to the hardware, and these differences must be taken into consideration when using VM2000. An overview of these constraints is provided below, with references to the relevant topic where necessary.

S servers exhibit the following deviations:

- The minimum main memory size and the granularity of the memory allocation is 1 Mbyte (see [page 265](#)).
- The main memory must be at least 64 Mbytes in size for the monitor VM.

SQ servers exhibit the following differences:

- Hot spare CPUs are not available.
Cold spare CPUs can be configured, see [page 34](#).
- Channels and channel devices are not available.
- On SQ servers X2000 emulates devices with a virtual input/output path for the virtual machines. The functions for VM2000-global reconfiguration (in the BS2000 commands `/ATTACH-/DETACH-DEVICE` and `INCLUDE-/REMOVE-DEVICE-CONNECTION`) have no effect.
- Parallel Access Volumes (PAVs, channel device) are not available.
The inputs/outputs are automatically parallelized by X2000.
- The main memory is managed in multiples of 2 Mbytes (see [page 265](#)).
The main memory of a VM is no longer contiguous.
- The main memory should be at least 512 Mbytes in size for a VM.
- Global storage is not available.

2.5.3 Constraints that apply to VM2000

When operating a BS2000/OSD server with VM2000, various constraints and limitations have to be taken into account that do not apply in native operation. An overview of these constraints is provided below, with references to the relevant topic where necessary.

The following constraints apply to the hardware:

- For interactive operation in the guest system, the hardware connection conditions specified by BCAM must be met for each guest system (BCAM is the central component of the software product openNet Server). These restrictions can vary depending on the version of the guest system.
- The assignment of the peripherals to the VMs should be planned carefully to ensure that performance is not impaired (see [section “Notes on configuration for VM2000 operation” on page 358](#)).

The following constraints apply to the software:

- The parameters for VM2000 are set using the startup parameter service, the parameter file of the VM2-MON subsystem and (on S servers) the configuration file of the monitor VM (see [page 53](#)).
- A guest system cannot be started by an IPL device that is shared, i.e. in SHARED usage mode (see [page 156](#)).
- CPU times cannot be calculated as the difference between two STCK values under VM2000 (see [page 392](#)).
- The "Channel Subsystem Monitoring Facility" function (S servers) for recording detailed device-specific operating times via DCS using the performance measurement monitor openSM2 (monitoring program SERVICETIME) can only be enabled for **one** guest system. If the monitoring program SERVICETIME is started on another guest system, no measurement data is supplied for this guest system (see [page 392](#)).
- For shareable devices that are assigned to more than one VM (SH(I) usage mode), the Channel Subsystem Monitoring Facility (S servers) is **not** activated. The openSM2 performance measurement monitor does not receive any measurement data for these devices (see [page 392](#)).
- The BS2000 command /WRITE-IOCF of the IOCFCOPY subsystem (S servers) can only be executed in the monitor system.

The following software prerequisites may need to be satisfied:

- A prerequisite for working with job variables is the software product JV in the monitor system which is contained in the OSD/XC software package for SQ servers. See [page 148](#) and [page 272](#).
- If the additional mirror units or clone units of a pubset are to be used for the pubset assignment to a VM, the software product SHC-OSD is required in the monitor system, see [page 319](#).
- The software product SECOS is required for assigning privileges to a user task (which is not TSOS) in the monitor system, see [page 91](#).
- The software product OMNIS is recommended when the guest system is operated via \$VMCONS or logical consoles are used, see [page 112](#) and [page 101](#).

2.5.4 Constraints that apply to guest systems

When operating guest systems with VM2000, certain constraints compared with native operation of the guest systems must be taken into consideration. An overview of these constraints is provided below, with references to the relevant topic where necessary.

The following functions of VM2000 can only be used for the guest systems BS2000/OSD as of V7.0:

- Shutdown in the guest system (see [page 170](#))
- Main memory reconfiguration with memory sizes of 32 GB and above (see [page 270](#))
- Automatic adaption of assignment mode, usage mode and device attributes for new pubset devices (see [page 316](#))

2.6 Differences between S and SQ servers

Topic	S servers	SQ servers
Version of VM2000 current version	VM2000 V10.0	VM2000 V10.0
Hardware Architecture of the server	/390 architecture	X86-64 architecture
CPU Max. number of real CPUs of the server ¹	16	32
Max. number of real CPUs for VM2000 operation	16	16
Hot spare CPUs, managed by VM2000	For certain S servers	Not available; cold spare CPUs can be configured
Extra CPUs, managed by VM2000	For certain S servers	For certain SQ servers
Max. multiprocessor level of a VM ²	8	32
Peripherals Operated peripherals	Channel peripherals; FC peripherals via type FC channel	Bus and FC peripherals
Operated devices	Devices as of BS2000/OSD-BC V6.0	Devices as of X2000 V5.3 and OSD/XC V4.0
Dynamic I/O configuration modification	for Channel and FC devices and their connection via BS2000/OSD	for bus and FC devices and their connection via X2000
Storage media Main memory used	Real main memory of the server	Real main memory of the server
Smallest main memory unit managed	1 Mbyte	2 Mbyte
main memory size of the monitor VM	At least 64 Mbytes, Default: 64 MByte	At least 512 Mbytes, Default: 512 MByte
Global storage	Is supported	Not available
Monitor system Operating system	BS2000/OSD V8.0 or V9.0	BS2000/OSD V8.0 or V9.0 (in the OSD/XC software package)
IPL of the monitor system	Via SVP load screen	Via SQ manager or automatically at server startup
Restarting the monitor system via SVP	Via SVP load screen	Not available (IPL possible via SQ manager)

Topic	S servers	SQ servers
Guest systems Operating system	BS2000/OSD V6.0 up to V9.0 and later versions if released for VM2000	BS2000/OSD-BC V8.0 and V9.0 and later versions if released for VM2000
Max. number of VMs or guest systems ³	15	15
BS2000 consoles for operating the guest system	Via SKP 3970	Via SQ manager (KVP)
IPL device	Disk or tape device	Disk device
Installation of VM2000 Module/program libraries of VM2000 monitor and VM2000 hypervisor	SYSLNK... SYSPRG...	SKMLNK... (VM2000 monitor only)
VM2000 parameters	Startup parameter file and parameter file of the VM2-MON subsystem	Startup parameter file and parameter file of the VM2-MON subsystem
Configuration of the monitor VM	Configuration file for the monitor VM in the home pubset of the monitor VM	Configuration file for the monitor VM in domain 0 (via the SQ manager)
Other settings	No	via the SQ manager

¹ The numbers in this line reflect the current release status when this manual went to print. VM2000 can manage up to 64 real CPUs.

² Defined by the firmware and the VM2000 implementation. Further restricted by the number of real normal CPUs in VM2000 operation, see [page 213](#)

³ VM2000 can manage up to 99 VMs.

3 Installing VM2000

You obtain the software product VM2000 as part of a software delivery via the SOLIS delivery system. The installation of VM2000 is carried out using the IMON installation monitor, see the “IMON” [13] manual.

VM2000 consists of the release units VM2000-MON (VM2000 monitor) and VM2000-UTIL (VMDUMP).

Ⓢ For S servers VM2000 also contains the release unit VM2000-HPV (VM2000 hypervisor).

The files of the release units VM2000-MON and VM2000-HPV are installed on the TSOS user ID by IMON. Following installation they can be moved manually from TSOS to another user ID. The new file names may then need to be specified as fully qualified names (e.g. in the VM2000 parameters).

For the release unit VM2000-UTIL (VMDUMP) the user ID can be selected when installation takes place.

The application of VM2000 must be prepared following installation using IMON. Details of this are provided in the sections below.

The release units of VM2000 have the following release items:

Component	Meaning	Standard file name
Subsystem catalog for VM2000 monitor	Contains subsystem declarations for the VM2-MON (monitor) subsystem	SYSSSC.VM2000-MON.<ver> ¹
Object module library for VM2000 monitor	Contains the object modules for the VM2-MON subsystem (hardware dependent)	SYSLNK.VM2000-MON.<ver> SKMLNK.VM2000-MON.<ver>
RMS selectable unit ² for VM2000 monitor	Contains the Repls required for the VM2-MON subsystem	SYSRMS.VM2000-MON.<ver>
Message file for VM2000 monitor ³	Contains the messages for the VM2-MON subsystem	SYSMES.VM2000-MON.<ver>
SDF syntax file for VM2000 monitor	Contains the VM2000 command description for the VM2-MON subsystem	SYSSDF.VM2000-MON.<ver>
Parameter file for VM2000 monitor	Contains the parameters for the VM2-MON subsystem (sample file)	SYSPAR.VM2000-MON.<ver>.SSP

Component	Meaning	Standard file name
Program file for VM2000 hypervisor	Program file for the VM2000 hypervisor (S servers)	SYSPRG.VM2000-HPV.<ver>
RMS selectable unit VM2000 hypervisor	Contains the Reps required for the VM2000 hypervisor (S servers)	SYSRMS.VM2000-HPV.<ver>
Load library for VMDUMP	Contains the module for VMDUMP	SYSLNK.VM2000-UTIL.<ver>
Program file for VMDUMP	Program file for starting VMDUMP	SYSPRG.VM2000-UTIL.<ver>
Message file for VMDUMP	Contains the messages for VMDUMP	SYSMES.VM2000-UTIL.<ver>
RMS selectable unit for VMDUMP	Contains the Reps required for VMDUMP	SYSRMS.VM2000-UTIL.<ver>
ENTER file for VMDUMP	ENTER job for output of a VM2000 dump to file	SYSENT.VM2000-UTIL.<ver>
SYSSII files for VM2000	Structural and installation information files for installation with IMON	SYSSII.VM2000-MON.<ver> SYSSII.VM2000-HPV.<ver> SYSSII.VM2000-UTIL.<ver>
Release notice for VM2000	VM2000 release notice in German and English	SYSFGM.VM2000-MON.<ver>.D SYSFGM.VM2000-MON.<ver>.E

¹ <ver> stands for the version identifier, e.g. 100 for VM2000 V10.0

² The RMS delivery quantities (SYSRMS...) are already incorporated in the RMS depot when installation takes place using IMON. They are stored in separate REP files for the VM2000 hypervisor (S servers), VM2000 monitor and VMDUMP by RMS at loader setup.

³ The message file is already incorporated in the MIP parameter file at installation using IMON or subsequently using /MODIFY-MIP-PARAMETERS and activated by MIP. See also [page 51](#).

3.1 Generating and configuring the device peripherals

The same device peripherals can be operated in BS2000/OSD native operation and under VM2000.

3.1.1 Device peripherals on S servers

Hardware generation

When hardware generation takes place for the server using IOGEN, an IO configuration file (IOCF) is created which defines the entire IO configuration. The IOCF file created is transferred to the service processor in native mode or in the monitor system by means of `/WRITE-IOCF` and is available for the next system startup (in native mode or of VM2000). You can find detailed information on generating channel peripherals in the “System Installation” manual [1].



The BS2000 command `/WRITE-IOCF` of the IOCFCOPY subsystem can only be used in native operation or in the monitor system.

Redundant configuration

The devices should be configured in such a way as to ensure that their availability is not affected if a channel is (automatically) disconnected by the operating system, i.e. the devices should be made available across multiple channels and by different control systems.

Generating devices for guest system operation via KVP consoles

At hardware generation the device pairs that represent the KVPs set up in the SCP 3970 are generated with identical mnemonic device names (device type code 64, SKP2) and consecutive device addresses.

For information on the principle of operating guest systems via BS2000 consoles, see [page 100](#).

Generating virtual consoles

Devices that are supposed to be assigned to the guest systems as virtual consoles must be generated as physical consoles (device type code 02/03) in the ATTACHED state at hardware generation of the channel peripherals.

Generation takes place in the usual way together with other real devices on any available channel (of type 2 or of type S).

Detecting the device peripherals

The available device peripherals are detected dynamically by the VM2000 hypervisor, the monitor system and the guest systems from the data of the active IO configuration file (IOCF).

This ensures that VM2000, the monitor system and all of the guest systems know and manage the same device peripherals.

The configuration statuses for devices and connections in the monitor system and the BS2000 guest systems are preset or set via the startup parameter service (`IOCONF` parameter set), see the “Introductory Guide to Systems Support” [2].

Changes to the set of devices through a dynamic change of the I/O configuration (see [page 361](#)) are automatically taken into account by VM2000.

Handling unknown device type codes

If a guest system that has a version number higher than BS2000/OSD-BC V9.0 (successor version) offers new device type codes, these are supported in the monitor system by means of an existing standard device type code. You will find more information in the relevant release notice.

3.1.2 BS2000 devices on the SQ server

The real devices in the periphery are not directly visible for the monitor system and the other BS2000 guest systems. BS2000 guest systems know only the **BS2000 devices** emulated in X2000. This means that BS2000 devices connected to the SQ server are always X2000 emulations of real devices, channel connections being emulated for BS2000/OSD.

BS2000 devices, i.e. the emulated devices, are configured exclusively via the SQ manager in X2000, see the manual “SQ Series Business Servers, Operation and Administration” [18]. No hardware generation takes place in BS2000/OSD.

Device peripherals required

To prepare VM2000 operation, the devices which are to be provided for the VMs or guest systems must be configured.

A VM or its guest system requires at least:

- One KVP device pair for guest system operating. KVP devices are created implicitly by X2000 when a new KVP is configured.
- LAN devices (LOCLAN, ZASLAN, BRGLAN) for the communication system
- Disks for the home pubset

Detecting the device peripherals

The available BS2000 device peripherals are detected dynamically by the monitor system and the BS2000 guest systems from the configuration data of X2000. This ensures that VM2000, the monitor system and all of the BS2000 guest systems know and manage the same device peripherals.

The configuration statuses for devices and connections in the monitor system and the BS2000 guest systems are preset or set via the startup parameter service (IOCONF parameter set), see the “Introductory Guide to Systems Support” [2].

Changes to the set of devices through a dynamic change to the I/O configuration (see [page 361](#)) are automatically taken into account by VM2000.

Handling unknown device type codes

If a guest system that has a version number higher than BS2000/OSD-BC V9.0 (successor version) offers new device type codes, these are supported in the monitor system by means of an existing standard device type code. You will find more information in the relevant release notice.

3.2 Setting up the VM2000 monitor

The VM2000 monitor is designed as a separate DSSM subsystem VM2-MON with its own parameter file. It executes in the monitor system and implements the administration interface for VM2000 operation.



VM2000 is thus independent of the generation of BS2000/OSD.

The monitor REP file must be self-contained; it must not be incorporated in the BS2000 REP loader.

The VM2-MON subsystem has the following properties:

- it is independent of the operating system version in the monitor system
- it is loaded automatically by DSSM before `SYSTEM READY`
- it can process VM2000 commands from the `CMDFILE`
- it is active during the entire system run
- it cannot be controlled by DSSM commands
- it cannot be replaced while the system is running

The VM2000 monitor uses the functions available in BS2000/OSD, e.g.:

- administration from a privileged user task
- processing command input via SDF and passing it to the processing routines, which then convert these commands into hypervisor calls
- executing message output, guaranteed messages
- SHOW output in structured S variables
- accounting tasks for VM2000 and virtual machines.

The central VM2000 monitor functions are implemented by the following system tasks:

VM2M	Main task, command processing
VM2G	General task, executing or distributing jobs (S servers)
VM2J	Job variable task, updating the MONJVs of VM2000
VM2C	VMCONS task, execution of \$VMCONS
VM2P	Pubset task, determining the pubset devices
VM2E	Task for processing requests of the VM2000 Agent (SQ servers)

Parameter file for the VM2-MON subsystem

The parameters of the VM2-MON subsystem are determined in the parameter file SYSPAR.VM2000-MON.<ver>.SSP. The following parameters can be entered:

VMCONS-OPERID = VMADMIN / <name 1..8>

Operator ID of \$VMCONS (user ID, see [page 60](#)) for establishing a connection to \$CONSOLE.

VMCONS-OPERID-PASSWORD = *NONE / <c-string 1..8>

Password for the \$VMCONS operator ID. Hexadecimal passwords are not permitted.

VMCONS-OPERATOR-ROLE = VM2000 / <name 1..8>

Operator role for \$VMCONS (see [page 60](#)).

VMCONS-PASSWORD = *NONE / <c-string 1..4> / <x-string 1..8>

Password for establishing a connection to \$VMCONS. A password comprising only zeros (X'00...00') is ignored (the effect is the same as with *NONE).

The operand names and keywords must not be abbreviated. If no specification is made or if the specifications are errored, the defaults (underscored) are used. If parameters are specified more than once, the most recently specified value applies. Comment lines are introduced by an asterisk (*).

Coexistence of VM2000 V9.0 respectively V9.5 and V10.0 on BS2000/OSD-BC V8.0 respectively V9.0

Both VM2000 V9.0 respectively VM2000 V9.5 and VM2000 V10.0 can be installed on the home pubset of BS2000/OSD-BC V8.0 respectively V9.0 (for SQ servers in the software package OSD/XC).

The VM2000 parameter `VERSION=V09.0 / V09.5 / V10.0` controls the VM2000 version to be selected. The default is:

- `VERSION=V09.0` for BS2000/OSD V8.0 on S servers
- `VERSION=V09.5` for BS2000/OSD V8.0 on SQ servers
- `VERSION=V10.0` for BS2000/OSD V9.0

See [section “Setting the VM2000 parameters in the parameter service” on page 53](#).

Note on the coexistence of multiple VM2000 versions

If multiple versions of the subsystem VM2-MON are installed with IMON, the following must be taken into account:

- An entry for the version-specific message file is contained in the MIP parameter file for each installed version of VM2-MON. When the monitor system is started, the message files of all installed versions of VM2-MON are activated. In ongoing operation the message file which belongs to the started version of subsystem VM2-MON is used.
- An entry for the version-specific SDF syntax file is contained in the SDF parameter file for the latest version of VM2-MON installed. When the monitor system is started, both the SDF syntax file of VM2-MON which is entered in the SDF parameter file and the SDF syntax file which belongs to the started version of the subsystem VM2-MON are activated. In ongoing operation, the SDF syntax file which belongs to the started version of the subsystem VM2-MON is used.

The parameter files need not be cleared for VM2000 operation. If you want to prevent the superfluous message and syntax files from being activated, you can remove the superfluous message file from the MIP parameter file using `/MODIFY-MIP-PARAMETERS` and, if required, correct the SDF syntax file in the SDF parameter file using `/MODIFY-SDF-PARAMETERS`.

Migration from VM2000 V9.0 respectively V9.5 to VM2000 V10.0 on BS2000/OSD-BC V8.0

Migration from VM2000 V9.0 respectively V9.5 to VM2000 V10.0 is possible on a BS2000/OSD V8.0 monitor system without switching the monitor system.

See also above, “Coexistence of VM2000 ...”.

3.3 Installing the VM2000 message files

The message file `SYSMES.VM2000-MON.<ver>` of the VM2000 monitor is integrated into the MIP parameter file when VM2000 is installed using IMON; it is activated by MIP.

As an alternative to installation with IMON, the `SYSMES.VM2000-UTIL.<ver>` message file can also be incorporated in the startup parameter service by adding the message file to the MSGFIL system parameter in the SYSOPT-CLASS2 parameter set and activated when the monitor system is started:

```
MSGFILxx=<msg_filename>
MSGOFL=n
```

xx is the consecutive number (01..15) in the MSGFIL system parameter, and `msg_filename` is the fully qualified name of the message file.

Accordingly, the number `n` of the message files must be incremented by one.

Messages of VM2000

VM2000 messages have the message code `VMSxyyy`:

x specifies the message type:

- 0 messages from the VM2000 hypervisor (S servers)
- 1-4 messages from the VM2000 monitor
- 5 messages from VM2000 initialization when the monitor system is started up
- 6 messages from the VM2000 monitor especially for SQ servers
- 7-8 not used
- 9 special command return codes


yyy consecutive message number (3-digit, 000 to 999).

The “Meaning” and “Response” texts for a message can be requested by means of the BS2000 command `/HELP-MSG-INFORMATION VMSxyyy`.

You will find the messages using an HTML application on our manual server (URL: <http://manuals.ts.fujitsu.com>) instead of in the previous manual "System Messages" and on the DVD “BS2000/OSD SoftBooks”.


The guaranteed messages of VM2000 are listed in the related command descriptions, see also [section “Guaranteed messages” on page 424](#).

3.4 Installing the VM2000 hypervisor files (S servers)

 On SQ servers hypervisor functions are executed by the Xen hypervisor. It is not necessary to install the VM2000 hypervisor.

Installation of the **program file** and the **REP loader** for the VM2000 hypervisor is dependent on the startup mode in which the monitor system is to be initialized (see [page 65](#)). There are two options for installation:

- Initializing the monitor system with a dialog startup.
When a dialog startup is used, the file name of the program file (message VMS5005) and the REP file (message NSI0050) is requested at the BS2000 console. Therefore any name can be chosen for the VM2000 hypervisor files at startup of the monitor system.
- Initializing the monitor system with a fast or automatic startup.
In this case, the names of the VM2000 hypervisor files must be entered in the startup parameter file or the files must be created under their standard file names (SYSPRG.VM2000-HPV.<ver> or SYSREP.VM2000-HPV.<ver>) on the home pubset of the monitor system.

 The structure, loader setup, buffering and logging of the VM2000 hypervisor Reps are organized in the usual way, see the section entitled “System corrections” in the “Introductory Guide to Systems Support” [2].

The VM2000 hypervisor Rep file must be an independent Rep file; it must not be incorporated in other Rep loaders.

Current corrections for the VM2000 hypervisor can also be made online without interrupting operation. For more information, ask your service representative.

3.5 Setting the VM2000 parameters in the parameter service

The parameters of VM2000 are set in the startup parameter file of BS2000/OSD. The following parameters can be entered between the lines `/BEGIN VM2000` and `/EOF`:

`VERSION = V10.0 / Vmm.n[a]`

Version of VM2000, where `mm.n` is the version number and (for compatibility reasons) `a` is the correction status of the VM2000 version to be selected. The specification of a correction status is ignored.



In monitor system BS2000/OSD V8.0 the values `V09.0[a]`, `V09.5[a]` and `V10.0[a]` are accepted. In monitor system BS2000/OSD V9.0 the value `V10.0[a]` should be specified.

If this parameter is not specified for a BS2000/OSD-BC V8.0 monitor system, startup selects `VERSION=V09.0` (on S servers) or `V09.5` (on SQ servers).

If this parameter is not specified for a BS2000/OSD-BC V9.0 monitor system (for SQ servers in the software package OSD/XC), startup selects `VERSION=V10.0`.

 The following parameters are only relevant on S servers:

`CONFIGURATION = SYSPAR.VM2000-MON.<version> / <filename 1..54>`

Name of the configuration file for the monitor VM.



If this parameter is not specified, startup selects the default file of the VM2000 version (dependent on the `VERSION` parameter).

`HYPERVERSOR = SYSPRG.VM2000-HPV.<version> / <filename 1..54>`

Name of the VM2000 hypervisor file.



If this parameter is not specified, startup selects the default file of the VM2000 version (dependent on the `VERSION` parameter).

`HYPERVERSOR-SIZE = *BY-SYSTEM / <integer 4..255>`

Specifies the size of the hypervisor main memory in Mbytes (see [page 265](#)).

When `*BY-SYSTEM` is specified, VM2000 calculates the size of the main memory for the VM2000 hypervisor. The size is largely dependent on the number of devices generated.

You can also set the size of the hypervisor main memory explicitly with `<integer 4..255>` in order to arrive at a main memory size that is independent of the number of devices generated.



If the specified hypervisor main memory size is too small, VM2000 hypervisor bottlenecks can be expected at command input or when internal hypervisor jobs are processed. In this case, commands are terminated with message `VMS4513`, or the VM2000 administrator is informed at intervals (`VMS4514`).

HYPERVISOR-REP = SYSREP.VM2000-HPV.<version> / <filename 1..54>
Name of the VM2000 hypervisor's Rep file.



If this parameter is not specified, startup selects the default file of the VM2000 version (dependent on the `VERSION` parameter).

PATH-GROUP-ID = *COMPLEX / *FROM-SYSID
Specifies the format of the path group ID (see [page 342](#)).



This parameter is no longer significant as of VM2000 V8.0. For reasons of compatibility the parameter can still be specified.

The VM2000 parameter `PATH-GROUP-ID` is output in the `/SHOW-VM-RESOURCES` command with `INFORMATION=*CONFIGURATION`.

ASSIGN-BY-GUEST = *NO / *YES

This specification defines for all devices whether they must by default be assigned explicitly to a VM (`ASSIGN-BY-GUEST=*NO`) or whether they can also be assigned implicitly to a VM by the guest system (`ASSIGN-BY-GUEST=*YES`), see [page 288](#). In the case of `ASSIGN-BY-GUEST=*YES` the devices are assigned by VM2000 to the assignment set `*STDSET`. During operation these definitions can be changed for existing and new devices using `/MODIFY-VM-DEVICE-ATTRIBUTES`.

The settings for the VM2000 parameter `ASSIGN-BY-GUEST` are output in the `/SHOW-VM-RESOURCES` command when `INFORMATION=*CONFIGURATION` is specified.

The operand names and keywords must not be abbreviated.

If no specification is made or if the specifications are errored, the defaults (underscored) are used. If parameters are specified more than once, the most recently specified value applies. Comment lines are introduced by an asterisk (*). Refer to the "Introductory Guide to Systems Support" [2] for more details on the startup parameter service.

Example of a VM2000 parameter set (S server)

```
/BEGIN VM2000 _____ (1)
* VM2000 PARAMETERS
CONFIGURATION=$MYUSERID.CONF.VM2000 _____ (2)
HYPERVISOR=$MYUSERID.HYP.VM2000
HYPERVISOR-REP=$MYUSERID.HYPREP.VM2000
/EOF _____ (3)
```

- (1) Start of the VM2000 parameter set.
- (2) The settings for the configuration file of the monitor VM and the VM2000 hypervisor files are changed. The defaults are used for the remaining VM2000 parameters.
- (3) End of the VM2000 parameter set.



S2000/OSD offers automatic selection of the parameter file by startup. For the monitor system, the suffix from the IOCF generation (S servers) or the system name (SQ servers) is appended to the default parameter file. This also applies for guest systems under VM2000. If the guest system is started with `/START-VM`, the VM name is appended as a suffix. If the guest system on an SQ server is started via an automatic restart or via the SQ manager, the system name is appended as a suffix. You will find more information in the “System Installation” manual [1].

3.6 Installing the configuration file for the monitor VM (S servers)

On S servers, the attributes and resources for the monitor VM are defined in the configuration file.

- SQ** On SQ servers the attributes and resources for the monitor VM are set via the SQ manager, see [section “Settings for VM2000 in X2000 \(SQ servers\)” on page 62](#). Any existing configuration file on the home pubset is ignored on SQ servers.

This file is a SAM file, whose name is specified in the VM2000 parameters (default: SYSPAR.VM2000-MON.<ver>).


In the configuration file comment lines (/REMARK) are also permitted. Continuation lines are not permitted. A line should not exceed 159 characters in length.

The configuration file may contain the following VM2000 commands:

- /CREATE-VM (see command description on [page 452](#))

This command defines the following attributes and resources for the monitor VM:

- Main memory size of the monitor VM in Mbytes. The DIMENSION operand cannot be specified (MEMORY-SIZE = <integer 1..1048576>). Main memory size is at least 64 Mbytes (default value)
- minimum main memory size of the monitor VM
MIN-MEMORY-SIZE = *STD is recommended, Minimum: 64 Mbytes
The DIMENSION operand cannot be specified.
- VM name (VM-NAME)
(max. 8 characters); *default value*: VM0001
- CPU quota of the monitor VM (CPU-QUOTA)
(integer from 1 to 99), *default value*: 14
The CPU-QUOTA=*BY-VM-GROUP(. . .) operand cannot be specified as no VM group is created in the startup phase of the monitor system. The monitor VM can be added to a VM group later using /ADD-VM-TO-VM-GROUP.
- maximum CPU utilization of the monitor VM (MAX-CPU-UTILIZATION)
(integer from 1 to 100), *default value*: 100
- maximum IO utilization of the monitor VM (MAX-IO-UTILIZATION)
(integer from 1 to 100), *default value*: 100

- assigning the monitor VM to a CPU pool (CPU=POOL)
This operand cannot be specified.
The monitor VM is initially always assigned to the standard CPU pool. It can be assigned to a different CPU pool later using /ASSIGN-VM-TO-CPU-POOL.
- administration password (PASSWORD)
(max. 8 bytes); *default value*: *NONE (no password)
- Multiprocessor level (PROCESSOR = *MONO / *BI / *TRIPLE / *QUADRO / *OCTO / *MAXIMUM / *EXTRA-AND-NORMAL / <integer 1..8>),
Default value: *MAXIMUM, i.e. the monitor VM is created with the maximum possible number of virtual CPUs. This number corresponds to the number of real normal CPUs in the server, but must not be greater than 8 (S servers) virtual CPUs.
 If there is no productive operation in the monitor system but only administration, the multiprocessor level *MONO (with virtual spare CPU) or *BI (failsafe operation) should be selected to avoid performance losses in the VM2000 system.
- location of monitor VM in main memory (LOCATION)
This operand is of no significance.
The location of the monitor VM in main memory is determined by VM2000.
- VM index (VM=INDEX)
This operand is of no significance.
“1” is always assumed to be the VM index of the monitor VM.
- command/function range for the VM2000 administrator
(ADMIN-PRIVILEGES=*STD/*MIN); *default value*: *STD (full range)
- privileges of the monitor VM
(IO-RESET=*NO/*YES, IO-PRIORITY=*NO/*YES, AUTO-SNAP-ASSIGNMENT=*YES/*NO, ASSIGN-BY-GUEST=*NO/*YES without specification of PRIVILEGES=*PARAMETERS(. . .)).
For the meaning, see [page 145](#).
Assignment sets (ASSIGN-BY-GUEST=*YES) cannot be specified.
- specification whether, in the case of fixed CU assignment (dedicated CPUs), the monitor VM still retains control over a real CPU if the monitor VM's virtual CPU which runs on it is inactive (uninterruptible wait state).
(VM-ACTIVE-IDLE=*NO/*AT-DEDICATED-CPUS),
Default value: *NO (the real CPU is deleted)
- MONJV to monitor the VM (MONJV)
The operand cannot be specified for the monitor VM.

- `/ADD-VM-DEVICES UNITS=list-poss(28):<alphanum-name 2..2>/<x-text 4..4>`
(see command description on [page 428](#))

This command is used (however many times is necessary) to assign VM devices to the monitor VM. Wildcards, device ranges or pubsets cannot be specified here. The KVP console device pair and the devices of the home pubset are automatically assigned explicitly to the monitor VM at startup of the monitor system.

- `/ADD-VM-RESOURCES GS=*REAL,VM-IDENTIFICATION=1,VM-SHARED=*NO/*YES`
(see command description on [page 433](#))

This command is used to assign real global storage to the monitor VM for exclusive or shared use. This is the prerequisite for virtualizing global storage under VM2000.



Commands in the configuration file which do not refer to the monitor VM are ignored.

If `/CREATE-VM` or `/ADD-VM-RESOURCES` is issued more than once, the operand values of the most recent correct command are used to initialize the monitor VM.

If `/ADD-VM-DEVICES` is specified for a device more than once, the `TYPE` operand of the last correctly entered command is used for the device.

Example of a configuration file

```
/REMARK CONFIGURATION FILE (VM2000) _____ (1)
/CREATE-VM MEM-SIZE=128,CPU-QUOTA=35,VM-NAME=MONITOR _____ (2)
/ADD-VM-DEVICES VM-ID=1,UNITS=(C2,C3,Y0,Y1,H0,H1,H2,H3) _____ (3)
/ADD-VM-DEVICES VM-ID=1,UNITS=(XX,5000,501C),TYPE=*SD _____ (4)
/ADD-VM-RESOURCES GS=*REAL,VM-IDENTIFICATION=1,VM-SHARED=*NO _____ (5)
```

- (1) Comment line.
- (2) Set up monitor VM with 128 Mbytes main memory.
- (3) Certain devices are assigned exclusively to the monitor VM.
- (4) Some devices are assigned to be shared between the monitor VM and other VMs.
- (5) The real global storage is assigned to the monitor VM for exclusive use.

Initializing the monitor system **without** the configuration file is only possible if the **entire** paging area has been created on the home pubset of the monitor system.

Here the following cases must be distinguished:

- Dialog startup
In this case, it is possible to enter the commands listed above for the monitor VM. If this is not done, the default values apply to the monitor VM, and the KVP console and the devices of the home pubset are assigned to the monitor VM explicitly at startup.
- fast and automatic startup
In this case, the default values for `/CREATE-VM` apply (see [page 56](#)), and the KVP console and the devices of the home pubset are assigned to the monitor VM explicitly at startup.

3.7 Preparing to use \$VMCONS

The use of \$VMCONS for VM2000 operation is optional. If the recommended administration and operating media are used (see [section “VM2000 Operating strategy” on page 88](#)), the measures described in this section do not have to be carried out.

\$VMCONS connects to UCON via the system application \$CONSOLE. To operate correctly, \$VMCONS requires authorization to send VM2000 commands to UCON and to receive the corresponding messages from UCON. This means that it is necessary to define an operator ID and an operator role for \$VMCONS using BS2000 commands in the monitor system and to link these to each other.

1. Set up an operator ID for \$VMCONS used as the user ID in the user catalog of the home pubset (privilege: USER-ADMINISTRATION):

```
/ADD-USER USER-IDENTIFICATION=VMADMIN/<name 1..8>,
      PROTECTION-ATTRIBUTE=*PARAMETERS(
      LOGON-PASSWORD=*NONE/<<c-string 1..8>),
      ACCOUNT-ATTRIBUTES=*PARAMETERS(...)
```

The user ID must be specified as the parameter VMCONS-OPERID in the parameter file of the VM2-MON subsystem (default value: VMADMIN), see [page 49](#).

The password for the user ID must be specified as the parameter VMCONS-OPERID-PASSWORD in the parameter file of the VM2-MON subsystem (default value: *NONE). Hexadecimal passwords are not permitted.

If the software product SECOS is used, (see [\[10\]](#)) /MODIFY-LOGON-PROTECTION can be used to restrict access to VMCONS-OPERID to programs only (privilege: USER-ADMINISTRATION):

```
/MODIFY-LOGON-PROTECTION
      USER-IDENTIFICATION=VMADMIN/<name 1..8>,
      DIALOG-ACCESS=*NO,
      BATCH-ACCESS=*NO,
      RBATCH-ACCESS=*NO,
      OPERATOR-ACCESS-TERM=*NO,
      OPERATOR-ACCESS-PROG=*YES
```

2. Set up the operator role for \$VMCONS. This is a set of routing codes describing the range of tasks to be performed by \$VMCONS (privilege: SECURITY-ADMINISTRATION):

```
/CREATE-OPERATOR-ROLE OPERATOR-ROLE=VM2000/<name 1..8>,  
ROUTING-CODE=9, PUBSET=*HOME
```

The name of the operator role must be specified as the parameter `VMCONS-OPERATOR-ROLE` in the parameter file of the VM2-MON subsystem (default value: VM2000). Routing code “9” covers all VM2000 commands and messages.



To avoid undesirable reactions, use only routing code “9”.

3. Assign the \$VMCONS operator role to the \$VMCONS operator ID (privilege: SECURITY-ADMINISTRATION):

```
/MODIFY-OPERATOR-ATTRIBUTES USER-IDENTIFICATION=VMADMIN/<name 1..8>,  
ADD-OPERATOR-ROLE=VM2000/<name 1..8>,  
PUBSET=*HOME
```



To avoid undesirable reactions, do not assign any further operator roles to the operator ID.

3.8 Installing the guest systems

No special requirements need to be observed when installing these guest systems and the successor versions to BS2000/OSD V9.0, provided that these are released for use under VM2000 V10.0 (see Release Notices).

3.9 Settings for VM2000 in X2000 (SQ servers)

To operate VM2000, for SQ servers certain settings must be made in X2000, in addition to the settings in native operation.

X2000 is used for the hardware-related management of SQ servers.

3.9.1 Configuring VM2000

The configuration settings for VM2000 are specified via the SQ manager, see the manual “SQ Series Business Servers, Operation and Administration” [18].

Setting the operating mode

In the *BS2000 Operating Mode* tag of the *Server Unit Management* menu of the SQ manager the “VM2000” operating mode must be selected for VM2000 operation (instead of the “Native BS2000” operating mode). See [section “Initializing VM2000 operation \(SQ servers\)” on page 68](#).

Setting the attributes and resources for the monitor VM

These settings must be specified via the SQ manager before VM2000 operation is initialized for the first time.

The following startup settings can be specified or changed for the monitor VM in the *BS2000 Operating Mode* tag of the *Server Unit Management* menu:

- Multiprocessor level (number of virtual CPUs, default value: 1)
- Current, minimum and maximum main memory size in Mbyte (default value: 512)
- Devices assigned (default value: empty device list)
- Password for administration (default value: no password)

The monitor VM must be assigned (explicitly and exclusively) at least the IPL disk, the devices of the home pubset and a KVP device pair via the SQ manager. Devices for shared use with other VMs can also be assigned.

In VM2000 operation the default values are used for the attributes of the monitor VM which cannot be set via the SQ manager. Changes, for example in the `ASSIGN-BY-GUEST` attribute, must be made later using `/MODIFY-VM-ATTRIBUTES`.

Changes to the attributes and resources of the monitor VM in VM2000 operation have **no** influence on the startup settings of the monitor VM.

Settings for automatic IPL and shutting down the monitor system

The SQ manager can be used to specify settings for automatic IPL of the monitor system and the remaining runtime when the monitoring system is shut down.

The following settings for automatic IPL of the monitor system can be specified in the *Options* tab of the *Virtual Machines > BS2000 > monitor-vm* menu when the `MONITOR` domain is configured:

- IPL option (Automatic IPL: Yes/No).
- IPL device
- IPL console
- System name
- Startup mode



The IPL option is always reset (Automatic IPL: No) after the operating mode has been changed. The other settings for automatic IPL remain unchanged.

The remaining runtime for the BS2000 shutdown can also be specified here (default: 30 minutes). This time applies to the monitor system and all BS2000 guest systems when the server unit is shut down (see [page 79](#)).

3.9.2 Creating KVP consoles to operate the guest system

A VM or its guest system requires at least one KVP console for guest system operating.

The configuration of the KVP devices using the SQ manager is described in the manual “SQ Series Business Servers, Operation and Administration” [18].

3.9.3 Local and remote communication

There are several options, both in native operation and under VM2000, for configuring the communication connections of the VM or the guest system.

Information on configuring these connections via the SQ manager is provided in the manual “SQ Series Business Servers, Operation and Administration” [18].

Each communication device must be assigned to the relevant guest system. In addition, the communication paths in the relevant RDF files must be configured in all guest systems (BCAM) or the commands `/BCOPTION` and `/BCIN` must be issued for the connections between the guest systems.

LOCLAN

This port is emulated by X2000 and enables a LAN connection under VM2000 to the other VMs with BS2000 guest system which are also connected to LOCLAN, via TCP/IP. In addition, this establishes a connection to X2000 with a gateway of the communication components which are connected there.

The package size can be up to 64 Kbytes with LOCLAN.

ZASLAN

To connect a VM to external hosts, the Ethernet Controller integrated into the BS2000/OSD server can be used (TCP/IP and ISO-LAN connections via Ethernet).

The package size can be up to 1,500 bytes for ZASLAN.



An Ethernet Controller is shared by all guest systems connected to it. To do this, a separate ZASLAN connection must be configured in X2000 **for every VM**.

BRGLAN

Under VM2000 this connection to an internal LAN bridge enables a LAN connection using TCP/IP to other virtual machines which are also connected to the same LAN bridge. VMs with a BS2000 guest system (as with LOCLAN) and also XenVMs can be connected to the LAN bridge.

With BRGLAN, the packets can be up to 64 Kbytes in size.



The internal LAN bridge must be configured via the SQ manager. A separate BRGLAN connection must be configured in X2000 for each VM with a BS2000 guest system.

4 Initializing and terminating VM2000

This chapter describes the initialization and termination of VM2000 and the monitor system. In addition, notes are given on the loading, operating and termination of guest systems.

4.1 Initializing VM2000 operation

- S On S servers first the relevant firmware must be loaded and then a special hardware-dependent load procedure (IPL) must be executed when you change from Native operation to VM2000 operation (and vice versa).
- SQ On SQ servers the “VM2000” operating mode must be set (in place of the “Native BS2000” operating mode) using the SQ manager and then the monitor system must be started up.

4.1.1 IPL for S servers

The contents of the PARS field are extended in the PROGRAM LOAD FRAME (DETAIL-1) screen for system initialization in BS2000/OSD.

If the monitor system is to be loaded with VM2000, the number “1” must be entered left-justified in the PARS field (figure 5).

If the operating system is to be loaded without VM2000, the leftmost character must be a blank.

The LOAD-FUNCTION LOAD or START must then be performed.

```

----- PROGRAM LOAD FRAME: DETAIL-1 -----                                E81L01G

-LOAD-FUNCTION-                    -IPL-DEVICE
==>                                ==> 2
*1 START AUTO                      *1 PRESET GROUP -----+
*2 START FAST                       >*2 CURRENT GROUP -----+ |
*3 START DIALOG                     *3 UNIT ADDRESS  +-+ +
*4 START                             7803      7803      A108
*5 SYSTEM DUMP                       XXXX      XXXX
*6 LOAD CLEAR -----+              XXXX      XXXX
*7 LOAD NON CLEAR --+                XXXX      XXXX
|
+MT CONTROL-                         --- DETAIL-2 STATUS ---
PARMS => 1                            ==> 1
>*1 NL                               VM MODE : AVM/EX
*2 SL                               EXA MODE : ENABLE
*3 NL-REWIND                        IPL EXEC : ENABLE
*4 SL-REWIND
*ENTER EXECUTE
*PF3 GO TO BASIC FRAME
*PF9 GO TO DETAIL-2
    
```

Figure 5: “PROGRAM LOAD FRAME (DETAIL-1)” screen

To load the VM2000 firmware, VM mode must be set to **AVM/EX** in the PROGRAM LOAD FRAME (DETAIL-2) screen.

To load the native firmware, VM mode must be set to **NATIVE** in the PROGRAM LOAD FRAME (DETAIL-2) screen.

```

----- PROGRAM LOAD FRAME: DETAIL-2 -----                E81L01G

-VM MODE-                                -IPL EXECUTION-
  ==>  2                                ==>  1
  *1 NATIVE                              >*1 ENABLE
  >*2 AVM/EX                              *2 DISABLE

                                           *ENTER  EXECUTE
                                           *PF3    GO TO BASIC FRAME
                                           *PF9    GO TO DETAIL-1
    
```

Figure 6: "PROGRAM LOAD FRAME (DETAIL-2)" screen

4.1.2 Initializing VM2000 operation (SQ servers)

Operating modes

SQ servers have the operating modes “Native BS2000” and “VM2000”.

When an SQ server is started, the operating mode of the previous session is set again. The operating mode can also be changed during ongoing operation (i.e. the SQ server and any existing XenVMs execute, but BS2000 systems are terminated) via the SQ manager, see [page 62](#).



The operating mode applies (also when it is changed during ongoing operation) only for BS2000 guest systems.

When switching the operating mode from “Native BS2000” to “VM2000”, it may first be necessary to shut down the native BS2000 system of the domain `BS2000`.

Switching the operating mode to “VM2000” terminates the domain `BS2000` and a domain with the domain name `MONITOR` and the startup settings for the monitor VM is configured, see [page 62](#).

When switching the operating mode from “VM2000” to “Native BS2000”, first all BS2000 guests systems and then the monitor system must be terminated, see [section “Terminating the monitor system and VM2000” on page 79](#).

Switching the operating mode to “Native BS2000” terminates the domain `MONITOR` and a domain with the domain name `BS2000` and the startup settings for the native BS2000 system is configured.

Starting the monitor system

The monitor system can be started in various ways:

- Automatically when initializing the domain `MONITOR` (see [page 63](#))
- Manually via the SQ manager
- Using the SVP functions of the KVP console which is assigned to the monitor VM

See the manual “SQ Series Business Servers, Operation and Administration” [18].




4.2 Starting up the monitor system

The monitor system can be started up in DIALOG, FAST and AUTOMATIC mode (see the “Introductory Guide to Systems Support” [2]). When it is started up in VM2000 operation, special components of VM2000 also run.

4.2.1 Required components

Certain files and components are required for startup of the monitor system in addition to native operation. They can be created in the case of dialog startup under a file name of your own choice. In the other startup modes, they must be entered in the parameter file or set up under their default name.

The following files are required:

File / component	Function	Special for VM2000	Changed for VM2000	Unchanged like BS2000
Startup for BS2000/OSD	System initialization for BS2000/OSD and VM2000 (monitor system)			x
Organizational program	Organizational program of the monitor system, Standard-EXEC			x
Startup parameter file	Contains parameters for BS2000 and VM2000 operation (see page 53)		x	
Rep loader for BS2000/OSD-BC	Contains system corrections for BS2000/OSD-BC			x
 Hypervisor file for VM2000	Program file for the VM2000 hypervisor (see page 52) on S servers	x		
 Hypervisor Rep file	Contains system corrections for the VM2000 hypervisor on S servers	x		
 Configuration file for the monitor VM	Required on S servers if the default settings and the default device configuration of the monitor VM are to be changed (see page 56)	x		
MIP parameter file	Contains the message files			x
SDF parameter file	Contains the command files			x
Subsystem VM2-MON	Contains the components of the VM2000 monitor (including parameter syntax and message files)	x		


4.2.2 Startup procedure in the monitor system

Startup for the monitor system proceeds in the same way as in native operation of BS2000/OSD, apart from some additional actions for VM2000.

The “Introductory Guide to Systems Support” [2] provides a more detailed description.

- [S] On S servers the startup of the monitor system is terminated if initialization of VM2000 is faulty.
- [SQ] On SQ servers startup of the monitor system is continued even if initialization of VM2000 is faulty. The error can then be corrected during ongoing operation and the monitor system can be restarted (e.g. by means of /SHUTDOWN and Restart).

If no parameter set for VM2000 is entered in the startup parameter file, the default values are used for the VM2000 parameters (see [page 53](#)).

-  Startup checks whether a version specified in the parameter file matches the loaded version of the VM2000 monitor (VM2-MON subsystem) and the VM2000 hypervisor (S servers).

The attributes and resources of the monitor VM were set when VM2000 was installed, see [chapter “Installing VM2000” on page 43](#).

- [S] Regardless of what is specified for the monitor system in the PROCESSOR operand, **all** of the real (normal) CPUs that are ready for operation are started at startup of VM2000. The hot spare CPUs and extra CPUs are also made ready for operation.

System initialization for the monitor system is terminated with the messages:

```
VMS5541 *** VIRTUAL MACHINE SYSTEM VM2000 V10.0A READY
          (COPYRIGHT (C) FUJITSU TECHNOLOGY SOLUTIONS 2011
          ALL RIGHTS RESERVED) ***
NSI0000 *** S Y S T E M   R E A D Y   ***
```

Dialog startup

In the case of a dialog startup, after the corrections for the operating system have been requested, entries are requested on S servers for the VM2000 hypervisor and the monitor VM.

- [SQ] On SQ servers the attributes and resources for the monitor VM are set via the SQ manager, see [section “Settings for VM2000 in X2000 \(SQ servers\)” on page 62](#). No further specifications can be made for VM2000 when startup takes place.

The entries can either be contained in the configuration file or be input directly at the BS2000 console. After further entries for the VM2000 hypervisor, the dialog startup procedure is continued in the same way as for native operation.

The steps of the dialog startup procedure relevant for VM2000 are outlined below:

- The operator is requested to enter the file name of the VM2000 hypervisor file:

```
VMS5005 ENTER VM2000 HYPERVISOR FILE NAME.
      REPLY(<FILE NAME>; EOT(USE STANDARD FILE))
```

The following responses are possible:

P. A VM2000 hypervisor file which is cataloged in the parameter file or under the standard file name `SYSPRG.VM2000-HPV.<ver>` is loaded.

P.<filename> A VM2000 hypervisor file which is created as a PAM file under the name <filename> is loaded (see [page 52](#)).

- The operator is requested to enter the file name of the VM2000 hypervisor Rep file:

```
NSI0050 SPECIFY VM2000 HYPERVISORREP FILE OR DEVICE.
      REPLY ( EOT ( USE STANDARD FILE ); FN=FILENAME, ( VOL=VSN ); CONS; END )
```

The following responses are possible:

P. The standard VM2000 hypervisor Rep file entered in the parameter file or cataloged under the standard file name `SYSREP.VM2000-HPV.<ver>` is processed (see [page 52](#)).

P.FN=<filename>

A VM2000 hypervisor Rep file created as a SAM file under the file name <filename> is processed.

P.CONNS Direct entry of Repls at the BS2000 console. The dialog is the same as when entering Repls in native operation.

P.END Entry of the Repls is terminated. Startup continues.



More than one file can be processed. Additional Repls can be entered at the BS2000 console.

- The operator is requested to input the file name of the configuration file for the monitor system or to input VM2000 commands at the BS2000 console.

```
VMS5027 SPECIFY VM2000 CONFIGURATION FILE NAME.
      REPLY (EOT(USE STANDARD FILE); FN=<FILE NAME>(,VOL=<VSN>);
      CONS(CONSOLE CMD INPUT); END(END OF INPUT))
```

The following responses are possible:

P. A configuration file of the monitor VM entered in the parameter file or cataloged under the standard file name `SYSPAR.VM2000-MON.<ver>` is processed.

P.FN=<filename>

The configuration file <filename> is processed.

P.CONSOLE Direct input of VM2000 commands at the BS2000 console. The VM2000 commands are requested with the message:

```
VMS5021 ISSUE VM2000 COMMAND (MAX. 70 CHARACTERS)
      OR 'END' (EOT=END). REPLY (DESIRED INPUT)
```

One or more VM2000 commands `/CREATE-VM`, `/ADD-VM-DEVICES` and `/ADD-VM-RESOURCES` can then be specified for the monitor VM (see [page 56](#)).

The maximum permissible input length is 70 characters (without P.).

P. terminates the input of the VM2000 commands from the BS2000 console.

P.END Input for VM2000 is terminated. Startup continues.

- The message VMS5027 is repeated until the response P.END is received. Several configuration files can be processed. The last `/CREATE-VM` correctly entered overwrites the previous `/CREATE-VM`.
- If P.END is entered immediately in response to message VMS5027, the default values are assumed for the attributes of the monitor VM (see [page 56](#)).
- Errors in the configuration file commands are reported at the BS2000 console. File processing continues.

4.2.3 Example of startup in the monitor system

This example shows a startup of the monitor system BS2000/OSD V9.0 on an S server. The output shown omits the parts which do not affect VM2000 (indicated by "...").

```

NSI00E3 IPL-REPS READ: 0; EXECUTED: 0
NSI1100 IPL DEVICE = WUV0.0; IPL PATH = 50A4 (MN=50A4)
NSI1106 *** BS2000 DIALOGUE STARTUP ***
NSI1110 ENTER OPTIONS OR EOT. REPLY (UNLOCK,TEST,ALLDISK,DRV-SELECT,
      CREATE-DRV,IPL-CONF)

p.
NSI1143 003 PUBVOLS OF HOME-PVS WUV0 WITH SYSID 160 ONLINE
NSI1145 WUV0.0 2012-01-19 16:11:40 ON 50A4
NSI1145 WUV0.1 2012-01-19 16:11:40 ON 50A5
NSI1145 WUV0.2 2012-01-19 16:11:40 ON 50A6
...
NSI1153 STATE OF PROCESSORS ONLINE:
NSI1155 CPU 00 ONLINE, ATTACHED (IPL CPU)
NSI1158 CPU 01 ONLINE, DETACHED
NSI1163 LOCAL DATE = 2012-01-19, TIME = 16:15:12 FROM SVP
NSI1180 LOAD ADDRESS OF SYSPRG.STRT.180 = 00767000
NSI00E6 SYSREP.STRT.180 IS USED AS REP FILENAME
NSI0050 SPECIFY PARAM FILE OR DEVICE.
      REPLY (EOT (USE STANDARD FILE);FN=FILENAME,(VOL=VSN);CONS;END)

p.
NSI00E6 SYSPAR.BS2.180 IS USED AS PARAM FILENAME
NSI008A PROCESSING PARAMETER ADD FILE PARAMS.ACCOUNT
...
NSI008A PROCESSING PARAMETER ADD FILE PARAMS.VM2000 _____ (1)
NSI0050 SPECIFY PARAM FILE OR DEVICE.
      REPLY (EOT (USE STANDARD FILE);FN=FILENAME,(VOL=VSN);CONS;END)

p.end
NSI1190 ENTER BS2000-FILENAME.
      REPLY (FILENAME(,VOL=VSN); EOT (USE STANDARD FILE))

p.
NSI0050 SPECIFY BS2000 REP FILE OR DEVICE.
      REPLY (EOT (USE STANDARD FILE);FN=FILENAME,(VOL=VSN);CONS;END)

p.
NSI00E6 SYSREP.BS2.180 IS USED AS REP FILENAME
NSI0050 SPECIFY BS2000 REP FILE OR DEVICE.
      REPLY (EOT (END); FN=FILENAME,(VOL=VSN);CONS;END)

p.end _____ (2)

```

(1) The VM2000 parameters are also read in the startup parameter service.

(2) The Rep entry for BS2000/OSD is terminated.

```

NSI0028 AVAILABLE MEMORY = 24.0 GB. CPU = 10020001 22600000
NSI5103 BS2000 : NAME = 012BXS, VERSION = 18.0A00,
      GEN-TIME = 2011-11-10 13:38:29
NSI5102 COPYRIGHT (C) FUJITSU TECHNOLOGY SOLUTIONS 2009
      ALL RIGHTS RESERVED
VMS5005 ENTER VM2000 HYPERVISOR FILE NAME.
      REPLY (<FILE NAME>; EOT(USE STANDARD FILE))

p. _____ (3)
VMS5015 VM2000 HYPERVISOR LOADED FROM FILE 'SYSPRG.VM2000-HPV.100'
NSI0050 SPECIFY VM2000 HYPERVISORREP FILE OR DEVICE.
      REPLY (EOT (USE STANDARD FILE);FN=FILENAME,(VOL=VSN);CONS;END)

p. _____ (4)
NSI00E6 SYSREP.VM2000-HPV.100 IS USED AS REP FILENAME
NSI0050 SPECIFY VM2000 HYPERVISORREP FILE OR DEVICE.
      REPLY (EOT (END); FN=FILENAME,(VOL=VSN);CONS;END)

p.end _____ (5)
VMS5027 SPECIFY VM2000 CONFIGURATION FILE NAME.
      REPLY (EOT(USE STANDARD FILE); FN=<FILE NAME>(,VOL=<VSN>);
      CONS(CONSOLE CMD INPUT); END(END OF INPUT)) _____ (6)

p. _____ (7)
VMS5014 FILE 'SYSPAR.VM2000-MON.100' IS USED AS VM2000 CONFIGURATION FILE
/CREATE-VMVM-I=1,VM-NAME=MONITOR,CPU-Q=9,MEM-S=320,PASS=P,PRO=BI,MIN-MEM=256
/ADD-VM-DEV (C2,C3,C4,C5,B492,B493,B494,B495,B470),1 _____ (8)
/ADD-VM-DEV (RK,RL,PO,P1),1
/ADD-VM-DEV (CO),1
/REMARK ***** XCS DISKS *****
/ADD-VM-DEV (6B30,6B31),1,SD
/ADD-VM-DEV MA,1
/REMARK EMULATED ZAS-LAN OF SKP
/ADD-VM-DEV (WO,W1),1
/REMARK EMULATED ZAS-LAN OF BACKUP-SKP
/ADD-VM-DEV (YO,Y1),1
/REMARK GATEKEEPER EMC39
/ADD-VM-DEV (B47A,B47B,B47C,B47D,B47E,B47F),1,SD

```

- (3) The default hypervisor file is to be used.
- (4) The default VM2000 hypervisor Rep file is to be used.
- (5) The input for the VM2000 hypervisor Rep file is concluded.
- (6) Input is requested for VM2000. It is possible to specify the configuration file for the monitor VM or enter VM2000 commands for the monitor VM on the BS2000 console.
- (7) The default configuration file is to be used.
- (8) The commands of the configuration file are logged.

```

VMS5027 SPECIFY VM2000 CONFIGURATION FILE NAME.
      REPLY (EOT(USE STANDARD FILE); FN=<FILE NAME>(,VOL=<VSN>);
      CONS(CONSOLE CMD INPUT); END(END OF INPUT))
p.cons _____ (9)
VMS5021 ISSUE VM2000 COMMAND (MAX. 70 CHARACTERS) OR 'END'.
      REPLY (DESIRED INPUT; EOT=END) _____ (10)
p.add-vm-devices (83fc,83fd,83fe,83ff),1 _____ (11)
VMS5021 ISSUE VM2000 COMMAND (MAX. 70 CHARACTERS) OR 'END'.
      REPLY (DESIRED INPUT; EOT=END) _____ (12)
p. _____ (12)
VMS5027 SPECIFY VM2000 CONFIGURATION FILE NAME.
      REPLY (EOT(USE STANDARD FILE); FN=<FILE NAME>(,VOL=<VSN>);
      CONS(CONSOLE CMD INPUT); END(END OF INPUT))
p.end _____ (13)
VMS5049 NUMBER OF REAL CPU(S) RUNNING: 2 _____ (14)
VMS5050 NUMBER OF AVAILABLE SPARE-CPU(S): 1
VMS5053 NUMBER OF AVAILABLE EXTRA-CPU(S): 0
VMS5020 STARTUP OF THE MONITOR SYSTEM IN BIProcessor MODE _____ (15)
VMS5016 TOTAL MAIN MEMORY FOR VM2000: 24539 MB;
      MONITOR VM SIZE: 320 MB; HYPERVISOR SIZE: 35 MB _____ (16)
EMM2960 EFFECTIVE SYSSIZE VALUE IS 240 MB
NSI5150 CPU'S ONLINE: 2 CPU'S INSTALLED: 2
ETMGT10 GTIME-VALUES FOR SESSION: ZONE=+01:00, DIFF=1:00
ETMGT01 TODR-EPOCH FOR SESSION: EPOCH=00
NSI3130 IPL-CONF SAVED ONTO DISK MN=50A4 WITHOUT ERRORS
NSI5104 IOCF: NAME = S2100001S2100001 / STANDARD OSDV3-8 / 11.01.12
      GEN-TIME = 2012-01-11 14:59:09
  
```

- (9) VM2000 commands are to be entered for the monitor VM on the BS2000 console.
- (10) A VM2000 command is requested. /CREATE-VM, and /ADD-VM-DEVICES and /ADD-VM-RESOURCES are permitted for S servers.
- (11) The monitor VM is assigned further devices with /ADD-VM-DEVICES. The maximum permissible input length for the command is only 70 characters here (not counting P.).
- (12) The entry of VM2000 commands on the BS2000 console is to be terminated.
- (13) Startup is to be continued.
- (14) Outputs the number of real normal CPUs which are available for VM2000 operation In addition, an spare CPU is available.
- (15) The monitor VM is operated as a biprocessor system.
- (16) VM2000 logs how much main memory is currently available.

```

EMM2301 PAGING AREA ON VOLUME WUV0.0 SIZE = 00400.00 MB.
EMM2301 PAGING AREA ON VOLUME WUV0.1 SIZE = 00400.00 MB.
EMM2301 PAGING AREA ON VOLUME WUV0.2 SIZE = 00400.00 MB.
EMM2850 THE SIZE OF THE PAGING-AREA IS 0001200.00 MB;
        THE RESERVED SIZE OF THE PAGING-AREA IS 0000011.30 MB.
BS2000 EXECUTIVE: CLASS2 ACTIVE
...
NMH1102 MESSAGE OUTPUT FILE ':WUV0:$TSOS.SYSMES.VM2000-UTIL.100',
        ACCESS=ISAM, ACTION=ADD _____ (17)
NMH1102 MESSAGE OUTPUT FILE ':WUV0:$TSOS.SYSMES.VM2000-MON.100',
        ACCESS=ISAM, ACTION=ADD
...
BLS0990 PROCESSING OF REP FILE ':WUV0:$TSOS.SYSREP.VM2000-MON.100' STARTED
...
CMD0695 SYSTEM SYNTAX FILE ':WUV0:$TSOS.SYSSDF.VM2000-MON.100' ACTIVATED
...
HLV0001 VMHEL FILE ':WUV0:$TSOS.SYS.VMHEL.2012-01-18.131824' OPENED
        WITH 'SPACE= 400' _____ (18)
VMS5541 *** VIRTUAL MACHINE SYSTEM VM2000 V10.0A READY
        (COPYRIGHT (C) FUJITSU TECHNOLOGY SOLUTIONS 2011.
        ALL RIGHTS RESERVED) *** _____ (19)
ESM0220 FUNCTION 'CREATE' FOR SUBSYSTEM 'VM2-MON /V10.0'
        COMPLETELY PROCESSED _____ (20)
ESM0501 'VM2-MON /V10.0' COPYRIGHT (C) FUJITSU TECHNOLOGY SOLUTIONS
        '2010'. ALL RIGHTS RESERVED
ESM0400 'CREATE' OR 'RESUME' SUBSYSTEM 'VM2-MON /V10.0' WITH
        'SYNCHRONOUS=*YES' AND 'RESET=*NO'
VMS2023 CPU 01 OF VIRTUAL MACHINE (1,MONITOR) STARTED _____ (21)
VMS2023 CPU 02 OF VIRTUAL MACHINE (1,MONITOR) STARTED _____ (22)
VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (1,MONITOR),
        CODE 80FFFFFF, CPU 02
VMS2011 CPU 02 OF VIRTUAL MACHINE (1,MONITOR) STOPPED
NSI0000 *** S Y S T E M   R E A D Y *** _____ (23)
...

```

- (17) The VM2-MON subsystem is loaded. In the process its message, REP and syntax files and also (without a message) its parameter file are processed.
- (18) The global VMHEL file is opened.
- (19) VM2000 is ready.
- (20) The VM2-MON subsystem has been loaded and is ready.
- (21) The second virtual CPU of the monitor VM was started by the monitor system.
- (22) The virtual spare CPU of the monitor VM was placed in operational status by the monitor system and then stopped again.
- (23) The system initialization of the monitor system is completed.

```

/dcs dcsof=sof.d017ze03 ----- (24)
...
VMS1030 BCAM APPLICATION '$VMCONS' READY ----- (25)
...
/show-vm-resources information=*configuration ----- (26)
MAX NUMBER OF VMS                15
MAX NUMBER OF VM-GROUPS          8
MAX NUMBER OF CPU-POOLS          2
MAX NUMBER OF CPUS PER VM        2
TOTAL REAL MEMORY SIZE           24539 MB
TOTAL MIN MEMORY SIZE            24539 MB
PATH-GROUP-ID MODE               COMPLEX
ASSIGNMENT SET   DEFAULT VALUE: *NONE
                  NEW DEVICES  : *NONE

CPU-ATTR    CPU-POOL    REAL CPUS
NORMAL      *STDPOOL    00(ATT),01(ATT)
EXTRA       NONE
SPARE       02(DET)
VMS3000 VM2000 COMMAND /SHOW-VM-RESOURCES FOR VIRTUAL MACHINE ( ,*VM2000)
          PROCESSED SUCCESSFULLY
NBR0740 COMMAND COMPLETED 'SH-VM-RESOURCES';
          (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: 2012-01-19
/show-vm-resources information=*all ----- (27)
          CPU-POOL(P)
VM-ID      VM-GROUP(G) CPU-Q  MAX-U  PROC  MEM-SIZE  MAX-IO
1 MONITOR  *STDPOOL(P)  9.00  100.00  2(+1)  320 MB  (100)

VM-ID      STATE          GUEST  VERSION ADMIN PRIV
1 MONITOR  RUNNING          BS2-OSD V09.0A NO  AS
F R E E   VM:14          LARGEST-DOMAIN: 24184 MB

VM-ID      CPU-POOL  GR-CPU-Q  CPU-Q/ EFF-Q  MAX-U/ EFF-U  PROC
1 MONITOR  *STDPOOL          9.00/100.00  100.00/100.00  2(+1)

VM-ID      SCHED ACT-I  VIRT CPUS
1 MONITOR  DC    NO    00(RUN),01(RUN),02(OFF)

```

(Output continued on the next page)

- (24) The data communication system is started in the server.
- (25) \$VMCONS is started automatically.
- (26) Command of the VM2000 administrator for outputting the VM2000 configuration information, see [page 601](#).
- (27) Command of the VM2000 administrator for outputting the status information on all VM2000 resources, see [page 591](#).

```

CPU-ATTR      CPU-POOL      REAL CPUS
NORMAL        *STDPOOL      00(ATT),01(ATT)
EXTRA
SPARE
                NONE
                02(DET)

VM-ID          VM-MONJV
                NO VM-MONJV ASSIGNED

VM-ID          UNITS
  1 MONITOR    ASSIGNED BY VM-ADMIN
                EXCL:  C0,C2,C3,C4,C5,MA,PO,P1,RK,RL,W0,W1,Y0,Y1,
                    50A4,50A5,50A6,83FC,83FD,83FE,83FF,B470,B492,
                    B493,B494,B495 _____ (28)
                SHARED: 6B30,6B31,B47A,B47B,B47C,B47D,B47E,B47F
DISPOSABLE    UNASS:  BA,BB,BC,BD,BE,BF,BG,BH,BI,BJ,BK,BL,BM,BN,BO,
                    BP,BQ,BR,BS,BV,BW,BX,BY,BZ,B0,B3,B4,B5,B6,B7,
                    B8,B9,CA(VC),CB(VC),CC(VC),CD(VC),CE(VC),CF,
                    CG,CH,CI,CJ,CK,CL,CM,CN,CO,CP,CQ,CR(VC),CS,
...
                2001,2002,2003,2004,2005,2006,2007,2008,2009,
                200A,200B,200C,200D,200E,200F,2010,2011,2012,
...
                FAF5,FAF6,FAF7,FAF8,FAF9,FAFA,FAFB,FAFC,FAFD,
                FAFE,FAFF
                SHARED: 6B30,6B31,B47A,B47B,B47C,B47D,B47E,B47F

VM-ID          PARTITION-ID ATTR      GS-SIZE      MODE      STATUS
                NO GS AVAILABLE

VM-ID          LOC-BEGIN  MEMORY-SIZE
HYPERVISOR    00000000      35 MB
  1 MONITOR    00000023      320 /      256 MB
NONE          00000163      24184 MB
LARGEST FREE  DOMAIN:      24184

VM-ID          PRIV      DETAILS
  1 MONITOR    AS

VMS3000 VM2000 COMMAND /SHOW-VM-RESOURCES FOR VIRTUAL MACHINE ( ,*VM2000)
          PROCESSED SUCCESSFULLY
NBR0740 COMMAND COMPLETED 'SH-VM-RESOURCES';
          (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: 2012-01-19

```

(28) In addition to the devices from the configuration file (see (7)), VM1 was also explicitly assigned the devices of the home pubset (50A4,50A5,50A6) when it was started, see [page 58](#).

4.3 Terminating the monitor system and VM2000

Because of the technical implementation of VM2000 (see [page 27](#)), termination of the monitor system has different effects on VM2000 and the server:

- S On S servers operation of VM2000 and of the BS2000/OSD server is terminated when the monitor system shuts down.
- SQ On SQ servers VM2000 operation is terminated when the monitor system shuts down (the XenVMs are not affected by this). The firmware component of the monitor system continues to run after the latter has shut down until the domain MONITOR has been terminated.

The server unit of the SQ server remains in operation even after the domain MONITOR has terminated; it can be terminated via the SQ manager.

A new IPL can subsequently take place on all server types.

Termination on S servers

VM2000 and server operation can be shut down on S servers in various ways:

- Using the VM2000 command `/SHUTDOWN-VM VM-ID=*VM2000(...)`.
 This is the recommended way to shut down VM2000, see [page 80](#) (coordinated shutdown of all guest systems and of the monitor system without restart)
- Using the BS2000 command `/SHUTDOWN` in the monitor system, see [page 83](#).
 As the guest systems are not shut down under VM2000 coordination here, the operator must take precautions to ensure correct system shutdown in the monitor system.

Termination on SQ servers

There are two reasons for regular termination of VM2000 operation on SQ servers:

- The server unit of the SQ server is to be shut down.
 Shutdown of the server unit is initiated via the SQ manager, see the manual “SQ Series Business Servers, Operation and Administration” [18]. Time-driven startup and shutdown of the server unit can also be configured via the SQ manager.

Every BS2000 guest system then receives a signal and shuts itself down, i.e. the shutdown action which is specified in the system parameter `SHUTPROC` of the guest system is initiated (default value: “immediate shutdown without a restart”).

When all BS2000 guest systems have been shut down or when half of the remaining runtime which is set has elapsed, the shutdown is initiated in the monitor system, i.e. the shutdown action defined there is executed. Alternatively, a coordinated shutdown of VM2000 can be executed beforehand, see [page 80](#).

Finally carrier system operation is terminated and the server unit is switched off.

- The operating mode “Native BS2000” is to be switched to (see [page 68](#)). In this case VM2000 operation must first be terminated (preferably by means of a coordinated shutdown of VM2000, see [page 80](#)). Only then can the operating mode be changed.

VM2000 operation is also terminated using the BS2000 command `/SHUTDOWN` in the monitor system (see [page 83](#)) or by shutting down the monitor system using the SQ manager. As the shutdown of the guest systems is not coordinated by VM2000 here, the operator must specify measures in the monitor system to ensure that VM2000 operation terminates correctly.

4.3.1 Coordinated shutdown of VM2000

The coordinated shutdown of VM2000 initially executes shutdown in all BS2000 guest systems as of BS2000/OSD V7.0. System shutdown can be initiated there by means of an enter job or by calling shutdown processing (similar to the BS2000 command `/SHUTDOWN`). The monitor system waits for system termination in the BS2000 guest systems. When all the BS2000 guest systems have been shut down or the wait time set has elapsed, the monitor system shuts itself down. In this case all VMs are also terminated. A detailed description of coordinated shutdown is contained in the [section “Shutdown in the guest system and of VM2000” on page 170](#).

An example of coordinated shutdown is provided on the following page.

In the case of coordinated shutdown, the following actions are performed by VM2000 when the monitor system is shut down:

- `/DELETE-VM` for all VMs which are still active, irrespective of the current system state (`CHECK-VM-STATE=*NO`)
- accounting for the monitor VM
- Detaching of the attached extra CPUs

Example of coordinated shutdown of VM2000

```

/show-vm-resources information=*std _____ (1)
CPU-POOL(P)
VM-ID      VM-GROUP(G)  CPU-Q   MAX-U   PROC   MEM-SIZE  MAX-IO
 1 MONITOR   *STDPOOL(P)  30.00  100.00  2(+1)   128 MB (100)
 2 SYSTEM2   *STDPOOL(P)  30.00  100.00  2(+1)   128 MB (100)
 3 SYSTEM3   *STDPOOL(P)  30.00  100.00  2(+1)   128 MB (100)

VM-ID      STATE          GUEST   VERSION ADMIN PRIV
 1 MONITOR   RUNNING        BS2-OSD V09.0A YES  AS
 2 SYSTEM2   RUNNING        BS2-OSD V09.0A NO   AS
 3 SYSTEM3   RUNNING        BS2-OSD V09.0A NO   AS
F R E E    VM:12          LARGEST-DOMAIN: 3686 MB

/shutdown-vm vm-id=*vm2000(waiting-time=*unlimited),
                action=*shutdown(quiet-time=2) _____ (2)
VMS2054 SHUTDOWN OF GUEST SYSTEM ON VM (2,SYSTEM2) INITIATED
VMS2054 SHUTDOWN OF GUEST SYSTEM ON VM (3,SYSTEM3) INITIATED
VMS2100 VM2000 SHUTDOWN INITIATED
***
*** Guest system operating of SYSTEM2 and SYSTEM3:
*** NRT1201 SHUTDOWN REQUESTED BY 'VM2000' _____ (3)
*** SHUTDOWN VIA VM2000 IN 02 MIN
*** ...
*** NRT1201 SHUTDOWN REQUESTED BY 'VM2000' _____ (4)
***

```

- (1) The command shows the initial status for this example: In addition to the monitor system, two further VMs are running with BS2000 guest system. The monitor VM is operated via the ADMIN dialog displayed.
- (2) The VM2000 administrator initiates the coordinated shutdown of VM2000. The monitor system would wait for an unlimited time for the guest systems to shut down. The QUIET-TIME for the guest systems (and the monitor system) is two minutes.
- (3) The guest system operating is informed about the impending shutdown and the duration of the QUIET-TIME. The message NRT1201 is output by the shutdown processing in the guest system when the shutdown is initiated (MODE=*QUIET).
- (4) After the QUIET-TIME has elapsed, the shutdown is initiated in the guest system (MODE=*END). The message NRT1201 is output again.

```

% VMS2051 GUEST SYSTEM ON VM (3,SYSTEM3) DOWN; REASON: SHUTDOWN
% VMS2051 GUEST SYSTEM ON VM (2,SYSTEM2) DOWN; REASON: SHUTDOWN  (5)
***
*** Guest system operating in the monitor system:
*** NRT1201 SHUTDOWN REQUESTED BY 'VM2000'  (6)
*** SHUTDOWN VIA VM2000 IN 02 MIN
*** ...
*** NRT1201 SHUTDOWN REQUESTED BY 'VM2000'  (7)
***
NRTT201 TASK TERMINATION DUE TO /SHUTDOWN COMMAND  (8)
EXC0419 /LOGOFF AT 1137 ON <date> FOR TSN 'OGLI'
EXC0421 CPU TIME USED: 9.2026

```

- (5) Guest systems SYSTEM2 and SYSTEM3 have been shut down. The VM2000 administrator is informed of this. Asynchronous messages are marked with % here.
- (6) The message NRT1201 is output by the shutdown processing in the monitor system when the shutdown is initiated (MODE=*QUIET) for the monitor system (and consequently for VM2000).
- (7) After the QUIET-TIME has elapsed, the shutdown is initiated in the monitor system (MODE=*END). The message NRT1201 is output again.
- (8) When the monitor system is shut down, the ADMIN dialog is also terminated.

4.3.2 Shutdown in the monitor system

VM2000 operation is also terminated when the monitor system is shut down using the BS2000 command `/SHUTDOWN` (see the “Introductory Guide to Systems Support” [2]) or (on SQ servers) by shutting down the monitor system via the SQ manager.

Before the monitor system is terminated, the operator should terminate each guest system by means of `/SHUTDOWN`. The operator in the monitor system must use the VM2000 command `/SHOW-VM-RESOURCES` to check if only the monitor system is actually still active.

At `/SHUTDOWN` of the monitor system with `MODE=*END` the following actions are also executed by VM2000:

- `/DELETE-VM` for all VMs which are still active, irrespective of the current system state (`CHECK-VM-STATE=*NO`)
- accounting for the monitor VM
- Detaching of the attached extra CPUs

In the case of `/SHUTDOWN` of the monitor system with `MODE=*END(RESTART=*YES(. . .))` or `MODE=*END,MESSAGE=C '*RESTART'` (with restart option), the guest systems and, on S servers, the VM2000 hypervisor remain active. The monitor system is reloaded. VM2000 performs the accounting for the monitor VM before the monitor system is restarted.

The guest systems are informed of the restart of the monitor system. Message NRTV002 (“Monitor system not ready. Restart has been initiated”) is output on the guest system BS2000 console. When the monitor system reaches `SYSTEM READY` again, message NRTV003 (“Monitor system ready again”).



In the case of `/SHUTDOWN` with restart the monitor system must be readied for automatic restart (see the “Introductory Guide to Systems Support” [2]).

In the event of an error (abnormal termination of the monitor system without a restart), VM2000 operation can be terminated by shutting down all the guest systems with `/SHUTDOWN` or (on SQ servers) by shutting down the BS2000 guest systems via the SQ manager.

4.4 Loading, operating and terminating guest systems

Note on the initialization of guest systems

- Loading (IPL) for a guest system is executed (after the VM has been configured) by means of the VM2000 command `/START-VM` (see [page 156](#)).
 - On SQ servers the guest system can also be loaded via the SQ manager or using the SVP functions of the KVP console which is assigned to the VM (see the manual “SQ Series Business Servers, Operation and Administration” [18]).

Notes on the operation of BS2000 guest systems

- The operation of the guest system on a VM (operating) normally takes place via KVP consoles (see [page 100](#)).

After `BCAM ACTIVE` in the guest system, logical consoles (e.g. via OMNIS) can be used with differentiated privileges for system operation in guest systems (see the “OMNIS/OMNIS-MENU” manual [11]).

A guest system can also be operated using `$VMCONS` (see [page 117](#)).

If the dialog system is ready for operation, operator commands can also be executed by user tasks with the OPERATING privilege.
- See [page 391](#) on local measurements in the guest system.

Notes on the termination of BS2000 guest systems

- The guest system is shut down using the VM2000 command `/SHUTDOWN-VM` (see [page 623](#)). As in native operation, it can also be shut down using the BS2000 command `/SHUTDOWN` (see the “Introductory Guide to Systems Support” [2]).
 - On SQ servers the shutdown of a guest system can also be initiated via the SQ manager.
- By means of the VM2000 command `/DELETE-VM . . . ,CHECK-VM-STATE=*NO` or `/SHUTDOWN` of the monitor system, the guest system is stopped regardless of the current state of the system. Files of the guest system may therefore be in an inconsistent state unless the guest system is terminated beforehand with `/SHUTDOWN`.

Information on status changes

The following status changes in the BS2000 guest system are indicated to the VM administrator by means of a VM2000 message:

- The guest system is ready for operation (VMS2050).
- The guest system has been terminated with /SHUTDOWN or has failed with SETS (VMS2051).
- The guest system is not ready for operation (/SHUTDOWN or SETS); a restart has been initialized (VMS2052).
- The active guest system was aborted with /RESET-VM or with /START-VM . . . ,CHECK-VM-STATE=*NO and restarted (VMS2051).

These messages are also distributed in the monitor system by means of routing code "9".

5 Operating VM2000 and the virtual machines (VMs)

This chapter describes the operating strategy of VM2000 and the functions of the VM2000 commands.

A detailed description of the syntax of the VM2000 commands and a description of the operands are given in [chapter “VM2000 commands” on page 415](#).

Operating levels on SQ servers

Actions for the VMs with a BS2000 guest system can be initiated on SQ servers from various operating levels:

- The full VM2000 functionality is available to the VM2000 and VM administrators via the user interface of VM2000 (which is described in this chapter). The VM2000 commands operate and manage the VMs with a BS2000 guest system, but not the XenVMs.
- The “start guest system”, “shut down guest system”, “abort VM” and “migrate VM” functions and information outputs are available to the administrators of the SQ server via the SQ manager (the user interface of the SQ servers) also for VMs with a BS2000 guest system.

5.1 VM2000 Operating strategy

The VM2000 operating strategy describes

- the three operating levels of VM2000:
 - operating the whole VM2000 system
 - operating a virtual machine
 - operating the guest system on a virtual machine
- operation of the virtual machines from a privileged user task
- the [Operating the guest systems using BS2000 consoles](#)
- the VM2000 administration application \$VMCONS:
 - establishing the connection to \$VMCONS
 - operating the virtual machines via \$VMCONS
 - operating the guest systems via \$VMCONS
- the options for automating VM2000 administration

5.1.1 Operating levels of VM2000

VM2000 has three operating levels. The [figure 7 on page 90](#) provides an overview of these.

Operating level whole VM2000 system (VM2000 administration)

The VM2000 system as a whole is operated by the VM2000 administrator. This is the privileged administrator who is permitted to enter all VM2000 commands (see [page 187](#)). The same person is also the VM administrator for the monitor VM.

The VM2000 administrator operates the VM2000 system and manages the resources of VM2000 in the administration dialog with the monitor VM, see [page 89](#)).

The whole VM2000 system can also be operated from a privileged user task in the monitor system (see [page 89](#)) or (e.g. when BCAM is not active in the monitor system) via BS2000 consoles in the monitor system (see [page 100](#)).

Operating level Virtual Machine (VM administration)

Each VM is operated by its VM administrator. The VM administrator is only permitted to enter VM administrator commands that affect that particular VM.

The VM administrator operates his or her VM in the administration dialog (see below).

The VM2000 administrator can also operate all of the VMs.

Administration dialog (known for short as the ADMIN dialog)

The dialog for operating the whole VM2000 system or for operating a single VM is referred to below as an **administration dialog (ADMIN dialog for short)**.



Recommendation

The ADMIN dialog is normally conducted from a privileged user task in the monitor system (see [page 91](#)).

The ADMIN dialog can also be conducted by means of the \$VMCONS administration application that runs in the monitor system (see [page 110](#) and [page 115](#)). The connection to \$VMCONS is established by means of an appropriate DCAM application. OMNIS is normally used for access via a terminal.

Operating level guest system on a VM (operating)

The guest system is normally controlled by the operator of the guest system by means of BS2000 consoles in the guest system (see [page 100](#)). See also the section entitled “Functions of the operator” in the “Introductory Guide to Systems Support” [2].

Operating from a privileged user task

If the dialog system is ready for operation, operator commands can also be executed parallel to the BS2000 consoles by user tasks with the OPERATING privilege:

- for operation in the monitor system and the guest systems
- For operating the VM2000 system as a whole by means of a user task with the OPERATING privilege in the monitor system (see [page 92](#)).

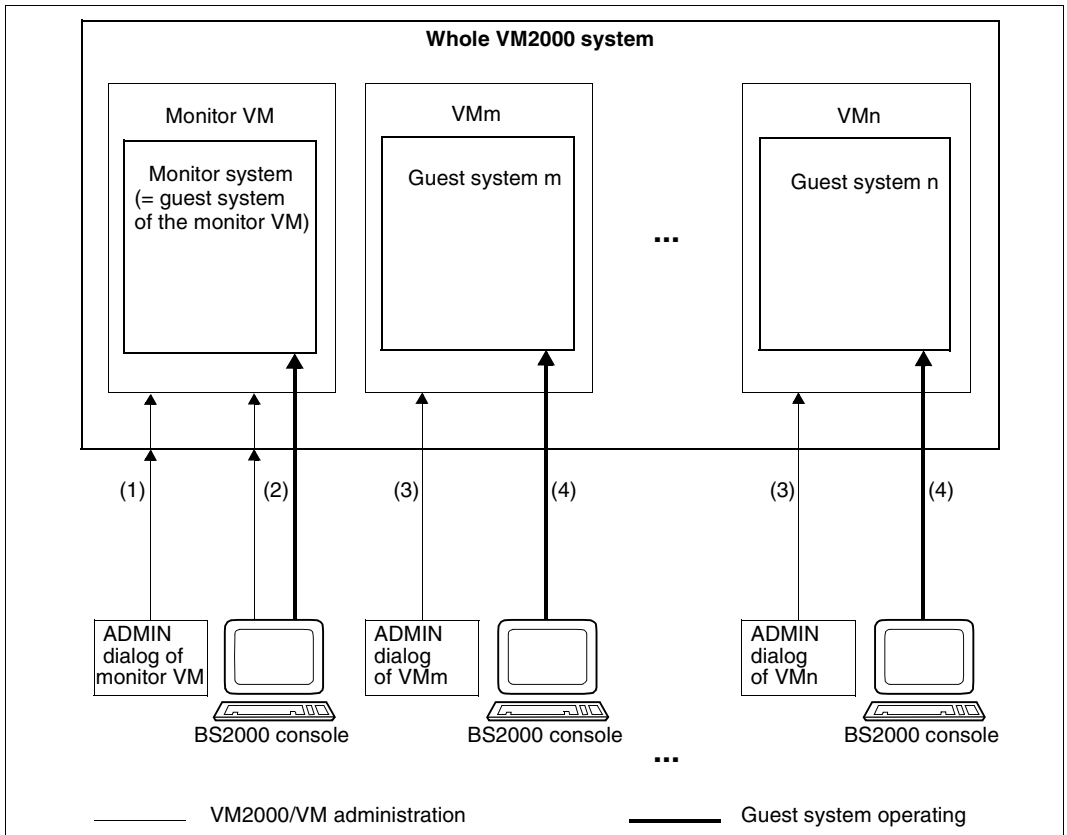


Figure 7: Operating strategy of VM2000 (recommendation)

- (1) All VM2000 commands can be entered via the ADMIN dialog of the monitor VM. The ADMIN dialog is conducted from a privileged user task in the monitor system.
- (2) The BS2000 operator commands for the monitor system and all VM2000 commands can be entered using BS2000 consoles in the monitor system.
- (3) The VM administrator commands for operating the associated VM can be entered via the ADMIN dialog of a VM. The ADMIN dialog is conducted from a privileged user task in the monitor system.
- (4) The operator operates the associated guest system on the VM using BS2000 consoles in the guest system.

5.1.2 Operating the virtual machines (ADMIN dialog from user task)

ADMIN dialogs can be conducted via privileged user tasks in the monitor system or via the administration application \$VMCONS which runs under the monitor system. An ADMIN dialog from a privileged user task offers the following advantages:

- mixing of BS2000 and VM2000 commands
- use of the full SDF functionality
- automation of operations through S and non-S procedures



The VM2000 command `/CALL-VM-PROCEDURE` is not available in a user task.

- storage of the procedure files under the user's own ID
- use of output in structured S variables (S-variable streams) (SHOW outputs, command return codes and guaranteed messages)
- receiving the asynchronous messages for the relevant administrator with the help of the event stream service (ESS)

5.1.2.1 Privileges of the user task in the monitor system

The privilege concept of SRPM increases security for system administration and simplifies task delimitation in the data center (see the "SECOS" manual [10]).

VIRTUAL-MACHINE-ADMINISTRATION privilege

A user task with the system-global privilege VIRTUAL-MACHINE-ADMINISTRATION is allowed to execute a subset of the VM2000 commands and thus to operate a virtual machine as the VM administrator.

The commands for the VM administrator and their demarcation from the VM2000 administrator commands are described in [section "Types of command" on page 187](#).

When the product is shipped, the privilege is assigned to the user ID TSOS. If SECOS is used, the security representative can assign the privilege to any other user ID (except his/her own).

VM2000-ADMINISTRATION privilege

A user task with the system-global privilege VM2000-ADMINISTRATION is allowed to execute all VM2000 commands and thus to operate the whole VM2000 system and all virtual machines as a VM2000 administrator.

The types of commands and their privileged functions are described in [section "Types of command" on page 187](#).

When the product is shipped, the privilege is assigned to the user ID TSOS. If SECOS is used, the security representative can assign the privilege to any other user ID (except his/her own).

OPERATING privilege

A user task with the system-global privilege OPERATING is allowed to execute all operator commands. In particular, it can also execute all VM2000 commands and thus operate the whole VM2000 system and all virtual machines as a VM2000 administrator.

When the product is shipped, the privilege is assigned to the user ID SYSOPR. If SECOS is used, the security representative can assign the privilege to any other user ID (except his/her own).

5.1.2.2 Administration dialogs from privileged user tasks

Operating a VM in the administration dialog (ADMIN dialog) from a privileged user task involves the following steps:

- ▶ Begin the BS2000 dialog under a user task with the appropriate privilege in the monitor system: `/SET-LOGON-PARAMETERS . . .`
- ▶ Open the ADMIN dialog using the VM2000 command

```
/BEGIN-VM-DIALOG VM-IDENTIFICATION=...,PASSWORD=...,TYPE=*STD/*ADMIN(...)
```

For the ADMIN dialog with the monitor VM, the VM2000-ADMINISTRATION privilege or the OPERATING privilege is required. If access to the VM is protected by a password, this must be specified when the dialog is opened.

After a successful `/BEGIN-VM-DIALOG`, the VM2000 administrator is informed of the task sequence number (TSN) of this ADMIN dialog (message VMS1013). The exchange of messages in the ADMIN dialog is also logged under this TSN in the CONSLOG file of the monitor system.

- ▶ Enter VM2000 commands to operate the VM (see [page 136](#) and [page 415](#)) or the VM2000 system as a whole (see [page 186](#) and [page 415](#)). BS2000 commands can also be entered.
- ▶ Terminate the ADMIN dialog with one of the VM2000 commands

```
/END-VM-DIALOG VM-IDENTIFICATION=...
/DELETE-VM VM-IDENTIFICATION=...
```

If `/DELETE-VM` is specified, VM2000 terminates all ADMIN dialogs with the VM and all VC dialogs with the guest system on the VM.



Under **one** user task, only **one** ADMIN dialog can be conducted at a time. For each VM up to 15 ADMIN dialogs from **different** user tasks can be conducted at the same time. Also, for each VM one ADMIN dialog can be conducted via \$VMCONS.

Logging

The VM2000 commands, outputs and messages for operating VM2000 and the VMs are logged in the CONSLOG file of the monitor system (see the “Introductory Guide to Systems Support” [2]).

The command inputs and message outputs belonging to an ADMIN dialog via a user task can be determined from the TSN, which is stored in the sender/recipient field of the CONSLOG entry. The VM2000 administrator is informed of the TSN at /BEGIN-VM-DIALOG via the message VMS1013.

In addition, the progress of the user task can be logged in the usual way to SYSLST (/SET-LOGON-PARAMETERS . . . , LOGGING=*PAR(LISTING=*YES)).

Using the event stream service (ESS)

The event stream of VM2000 is the system-dependent, asynchronous part of the data stream of a VM2000 or VM administrator dialog. It includes the periodic outputs of /SHOW-VM-STATUS and the asynchronous messages from VM2000, to which the administrator may have to respond.

The following **asynchronous messages** of VM2000 are normally received in the user event stream allocated to the privileged user task.

VMS0017	VMS0025	VMS0026	VMS0027	VMS1012	VMS1013	VMS1016	VMS1032
VMS1033	VMS1051	VMS1531	VMS1601	VMS1602	VMS2011	VMS2012	VMS2014
VMS2016	VMS2021	VMS2023	VMS2033	VMS2036	VMS2037	VMS2050	VMS2051
VMS2052	VMS2053	VMS2059	VMS4005	VMS4009	VMS4065	VMS4066	VMS4094
VMS4095	VMS4191	VMS4192	VMS4224	VMS4227	VMS4244	VMS4245	VMS4342
VMS4344	VMS4510	VMS4511	VMS4514	VMS4551	VMS4552	VMS4553	VMS4554
VMS4555	VMS4556	VMS4557	VMS4558	VMS4559	VMS4560	VMS4561	VMS4562
VMS4563	VMS4564	VMS4620	VMS4621	VMS6002	VMS6003	VMS6020	VMS6021
VMS6023	VMS6094						

With administration under a privileged user ID, you can receive the asynchronous messages for the relevant administrator with the help of the event stream service (ESS) and you can use the presentation functions of ESS.

The operating functions of ESS are also available if you have the OPERATING privilege. The ESS presentation functions are shown in the command description of /SHOW-SYSEVENT-LOG.

*Recommendation*

Use the event stream service to receive the event stream of VM2000. For examples of the structure, see the “Introductory Guide to Systems Support” [2], chapter “Operator functions”.

If you do not use the event stream service, the asynchronous messages will be output to SYSOUT at the next non-empty user input.

Even if you use the event stream service, (additional) output to SYSOUT is enabled (default). Output to SYSOUT can be suppressed by means of the task-specific command `/MODIFY-MSG-OPTIONS SYSTEM-MSG=*NO`.

5.1.2.3 Example of an ADMIN dialog from a user task

This example shows the interplay of BS2000 and VM2000 commands from a privileged user task on an S server.

```

./vm2adm set-logon-parameters user-id=tsos,logging=*par(listing=*yes) - (1)
/assign-syslst to=list.vm2adm _____ (2)
/begin-vm-dialog vm-id=1,password='moni' _____ (3)
/assign-sysevent to=*user-log _____ (4)
NBR3100 EVENT STREAM 'OBML' WITH TSN 'OBMM' ASSIGNED TO USER TASK 'OBML'
/modify-msg-options system-msg=*no _____ (5)
/show-vm-resources information=*std _____ (6)
CPU-POOL(P)
VM-ID      VM-GROUP(G)  CPU-Q    MAX-U    PROC    MEM-SIZE  MAX-IO
  1 MONITOR  *STDPOOL(P)  30.00   100.00   2(+1)   128 MB (100)

VM-ID      STATE          GUEST    VERSION  ADMIN  PRIV
  1 MONITOR  RUNNING        BS2-OSD  V09.0A  YES    AS
  F R E E  VM:14          LARGEST-DOMAIN: 3942 MB
/create-vm memory-size=128,vm-name=system2,cpu-quota=30,
           processor=*mono,vm-index=2,password=c'sys2' _____ (7)
VMS3050 VIRTUAL MACHINE (2,SYSTEM2) CREATED
VMS4615 VIRTUAL MACHINE (2,SYSTEM2) ASSIGNED TO CPU POOL '*STDPOOL'
/add-vm-devices units=e062,vm-id=system2,type=*sd _____ (8)
VMS4005 DEVICE 'E062' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2) (TYPE=SD)

```

- (1) Login under a privileged user ID in the monitor system (VM2000-ADMINISTRATION privilege). The process is to be logged.
- (2) A BS2000 command is entered to control the logging.
- (3) Login as VM2000 administrator by opening an ADMIN dialog with the monitor VM.
- (4) This BS2000 command assigns the user task a user event stream. Asynchronous messages from VM2000 to the VM2000 administrator are logged as of this point in the assigned user event stream.
- (5) The asynchronous messages are only to appear in the user event stream; they are not to be output additionally to SYSOUT.
- (6) The VM2000 administrator requests the status of VM2000.
- (7) A mono-VM is initialized by the VM2000 administrator. The VM receives the VM name SYSTEM2. The password SYS2 is defined for administration.
- (8) The VM2000 administrator assigns VM2 a disk device for shared use with other VMs.

```

/add-vm-dev units=(ch,ci),vm-id=system2 _____ (9)
VMS4005 DEVICE 'CH' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
VMS4005 DEVICE 'CI' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
/add-vm-devices vm-id=system2,
  units=*pubset-devices(pubset=*by-pubres-device(unit=e059)) _____ (10)
VMS4005 DEVICE 'E059' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
VMS4005 DEVICE 'E05A' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
VMS4005 DEVICE 'E05B' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
VMS4220 PUBSET '4IVW' WITH PUBRES DEVICE 'E059' ASSIGNED TO VM (2,SYSTEM2)
/start-vm ipl-unit=e059,information-byte=*fast,vm-id=system2 _____ (11)
/show-vm-attributes information=*all,vm-id=system2 _____ (12)
CPU-POOL(P)
VM-ID          VM-GROUP(G) CPU-Q    MAX-U    PROC    MEM-SIZE  MAX-I/O
  2 SYSTEM2    *STDPOOL(P) 30.00   100.00   1(+1)   128 MB (100) _____ (13)

VM-ID          STATE                GUEST  VERSION  ADMIN  PRIV
  2 SYSTEM2    RUNNING              UNKNOWN          NO    AS

VM-ID          CPU-POOL  GR-CPU-Q  CPU-Q/  EFF-Q  MAX-U/  EFF-U  PROC
  2 SYSTEM2    *STDPOOL                30.00/ 33.33 100.00/ 33.33 1(+1)

VM-ID          SCHED ACT-I  VIRT CPUS
  2 SYSTEM2    DC    NO    00(RUN),01(SLEEP) _____ (14)

VM-ID          VM-MONJV
  2 SYSTEM2    NO VM-MONJV ASSIGNED _____ (15)

```

(Output continued on the next page)

- (9) The VM2000 administrator assigns VM2 a KVP console device pair.
- (10) The VM2000 administrator assigns VM2 the disk device(s) of the home pubset by specifying the pubres device. The pubset is detected by VM2000.
- (11) The VM2000 administrator starts the guest system on VM2. The pubres device of the home pubset is selected as the IPL unit. A fast startup is to be executed. Operation of the guest system is carried out on a BS2000 console of the guest system.
- (12) The VM2000 administrator requests the attributes and status of VM2.
- (13) The attributes of the VM2 is output. The VM is running; a VM administrator is not connected. The guest system is not known because it is still in the startup phase.
- (14) The virtual CPU 00 of VM2 is running (fixed CPU assignment to a real CPU). The virtual spare CPU 01 is ready to operate.
- (15) No MONJV is assigned to the VM.


```

VM-ID          UNITS
2 SYSTEM2     ASSIGNED BY VM-ADMIN _____ (16)
              EXCL:  CH,CI,E059,E05A,E05B
              SHARED: E062

```

```

VM-ID          PARTITION-ID ATTR      GS-SIZE      MODE  STATUS
2 SYSTEM2     NO GS ASSIGNED _____ (17)

```

```

VM-ID          LOC-BEGIN  MEMORY-SIZE
2 SYSTEM2     0000008E      128 /      128 MB _____ (18)

```

```

VM-ID          PRIV  DETAILS
2 SYSTEM2     AS _____ (19)

```

```
/show-vm-attributes information=*std,vm-id=system2
```

```
CPU-POOL(P)
```

```

VM-ID          VM-GROUP(G) CPU-Q  MAX-U  PROC  MEM-SIZE  MAX-IO
2 SYSTEM2     *STDPOOL(P) 30.00 100.00 1(+1) 128 MB (100)

```

```

VM-ID          STATE          GUEST  VERSION ADMIN PRIV
2 SYSTEM2     RUNNING        BS2-OSD V09.0A NO  AS _____ (20)

```

```
/shutdown-vm vm-id=system2,action=*shutdown(quiet-time=0) _____ (21)
```

```
VMS2054 SHUTDOWN OF GUEST SYSTEM ON VM (2,SYSTEM2) INITIATED
```

- (16) List of the mnemonic device names that are explicitly assigned to the VM2. EXCL: indicates the devices that are assigned for exclusive use. SHARED: indicates the devices that are assigned for shared use with other VMs.
- (17) No global storage is assigned to the VM.
- (18) The size of the main memory for the VM and its position in main memory are output.
- (19) The VM has the (default) privilege AUTO-SNAP-ASSIGNMENT.
- (20) The guest system on VM2 has now reached SYSTEM READY. VM2000 recognizes that it is a BS2000 guest system.
- (21) The VM2000 administrator initiates the shutdown for the guest system on VM2 without a QUIET time.

```

/show-vm-attributes information=*std,vm-id=system2
          CPU-POOL(P)
VM-ID      VM-GROUP(G)  CPU-Q    MAX-U    PROC    MEM-SIZE  MAX-IO
  2 SYSTEM2 *STDPOOL(P)  30.00   100.00   1(+1)    128 MB   (100)

VM-ID      STATE          GUEST   VERSION ADMIN PRIV
  2 SYSTEM2 DOWN              BS2-OSD V09.0A NO   AS _____ (22)
    
```

```

/delete-vm vm-id=system2 _____ (23)
VMS4009 DEVICE 'CH' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4009 DEVICE 'CI' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4009 DEVICE 'E059' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4009 DEVICE 'E05A' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4009 DEVICE 'E05B' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4009 DEVICE 'E062' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4619 VIRTUAL MACHINE (2,SYSTEM2) REMOVED FROM CPU POOL '*STDPOOL'
VMS4024 VIRTUAL MACHINE (2,SYSTEM2) TERMINATED
    
```

/show-sysevent-log _____ (24)

```

File Edit View Help
-----
EVENT STREAM SERVICE
Log-Id: 0BMU User-Id: TSOS Owner-TSN: 0BMU Mode: Static
-----
Date: <date> Time: 10:21:36 Status: Active
Lines 1 thru 6 of 6
More:
% VMS2050 GUEST SYSTEM ON VM (2,SYSTEM2) READY
% VMS2033 'EXC0557 SHUTDOWN PROCESSING COMPLETED' FROM VM (2,SYSTEM2) VIA SVP
% VMS2051 GUEST SYSTEM ON VM (2,SYSTEM2) DOWN; REASON: SHUTDOWN
% VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (2,SYSTEM2), CODE
(80FFFFFF), CPU 00
*** End of actual presentation range ***

Command ==>
F1=Help F3=Exit F6=Dynamic F7=Backward F8=Forward F10=Menu F11=Index ...
    
```

- (22) After shutdown, VM2 is in the DOWN state.
- (23) VM2 is terminated by the VM2000 administrator.
- (24) The presentation function of the event stream service (ESS) is called. The asynchronous messages which have arrived for the VM2000 administrator are displayed.

```
/end-vm-dialog _____ (25)
/assign-sysevent to=*dummy _____ (26)
/assign-syslst to=*primary
/print-document from-file=list.vm2adm _____ (27)
/logoff system-output=*delete _____ (28)
```

- (25) The VM2000 administrator terminates his/her ADMIN dialog with the monitor VM.
- (26) The assignment of the user event stream to the user task is canceled.
- (27) The SYSLST assignment is canceled and the session log is printed out.
- (28) The user task is terminated and the output of SYSLST is suppressed.


5.1.3 Operating the guest systems using BS2000 consoles


In this manual the term **BS2000 console** is used when operating the monitor system or another guest system via a console screen is meant.

Where necessary, BS2000 consoles are subdivided into:

- KVP consoles

The operator obtains access to the operator functions of a guest system through the emulation of a console screen by the KVP (console distribution program).

 On S servers KVPs and their devices are managed and implemented by the SKP 3970, see [page 102](#).

 On SQ servers KVPs and their devices are managed and implemented via X2000, see [page 107](#).


- logical consoles (see [page 101](#))




You are recommended to operate the guest systems using KVP consoles or logical consoles.

- VC dialog via \$VMCONS (see [page 117](#))

This operating method is **no longer** recommended.

 On S servers the VC dialog is implemented by \$VMCONS via the “virtual console”.

 On SQ servers the VC dialog is implemented by \$VMCONS via a KVP connection.

Use of BS2000 consoles in VM2000

BS2000 consoles can be used in VM2000 operation:

- for IPL (SQ servers), startup and operating in the monitor system and in the other guest systems
- to operate the VM2000 system as a whole, e.g. when BCAM is not active in the monitor system



The authorization code “9” is required to operate the VM2000 system as a whole via a BS2000 console of the monitor system.

All VM2000 commands have the authorization code “9”.

All VM2000 messages have the routing code “9”.

Only VM2000 commands can be entered and only VM2000 messages can be received on a BS2000 console that **only** has the authorization code “9”.

A BS2000 console like this is therefore on a par with the administration dialog of the VM2000 administrator.

Operating guest systems via KVP consoles or logical consoles offers the following advantages compared to operating in a VC dialog:

- Direct communication with the guest system
- Guest system operation even if the data communication system is not available in the monitor system (e.g. in the case of an automatic restart of the monitor system).
- Uniform operation of the guest systems and the monitor system.

Logical consoles

After `BCAM ACTIVE` in the guest system, logical consoles with sophisticated options for privileges can be used for system operation in guest systems.

A logical console is a user program in which a DCAM application is running that communicates with the `$CONSOLE` system application. After `BCAM ACTIVE` it takes on console functions in a guest system (depending on the operator role).

Logical consoles can be set up, for example, using the software product OMNIS (see the "OMNIS/OMNIS-MENU" manual [11]). After a connection is set up to `$CONSOLE` (`OPNCON TYP=UCON`) with the specification of an operator ID, OMNIS controls the transfer of messages between the OMNIS terminal and `$CONSOLE`.

Logical consoles are connected via BCAM to the UCON task in the guest system. They can be used in VM2000 operation:

- for operation in the monitor system and the guest systems
After `BCAM ACTIVE`, all operation can be carried out via logical consoles, i.e. all authorization codes (routing codes) can be put on logical consoles in the guest system. All BS2000 commands for operating the guest system can be entered via logical consoles.
- for operating the VM2000 system as a whole by means of a logical console in the monitor system (authorization code "9").

5.1.3.1 KVP consoles on S servers (via SCP 3970)

The service/console processor SKP 3970 is an enhancement for S servers which is based on Linux (SKP 3970-40 and -50) or SINIX-2000 (SKP 3970-20), see the relevant SKP manual [20]. It supports operation, monitoring, diagnostics and maintenance for the S server, as well as remote service or Teleservice.

The so-called "SKP Manager" is the graphical user interface (GUI) of the SKP 3970-40 and -50. Operation of BS2000 from the SKP manager in native mode or as a VM2000 guest system is controlled by a console distribution program (KVP). The KVP parameters are created at SKP system installation and can be modified by means of configuration.

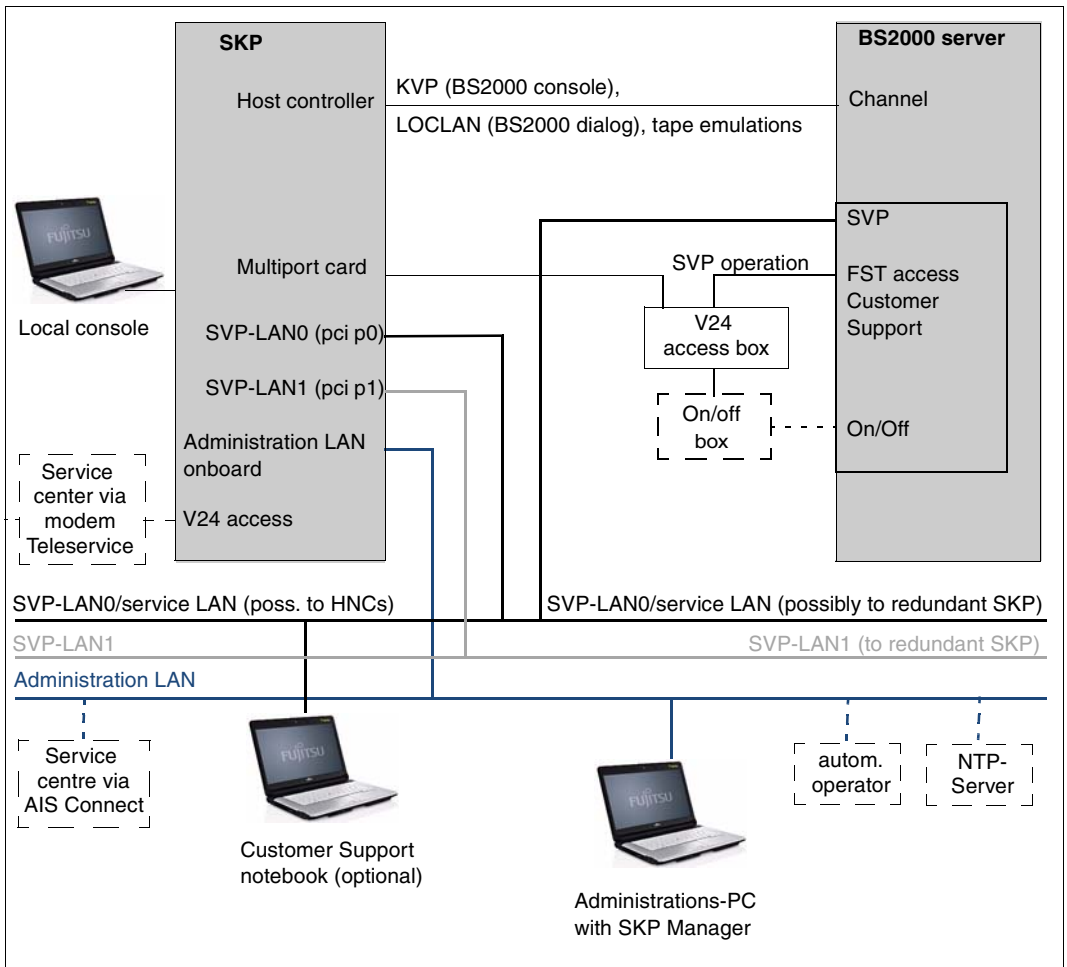


Figure 8: SCP 3970-40 and -50 (Linux) control concept with local and remote consoles (S servers)

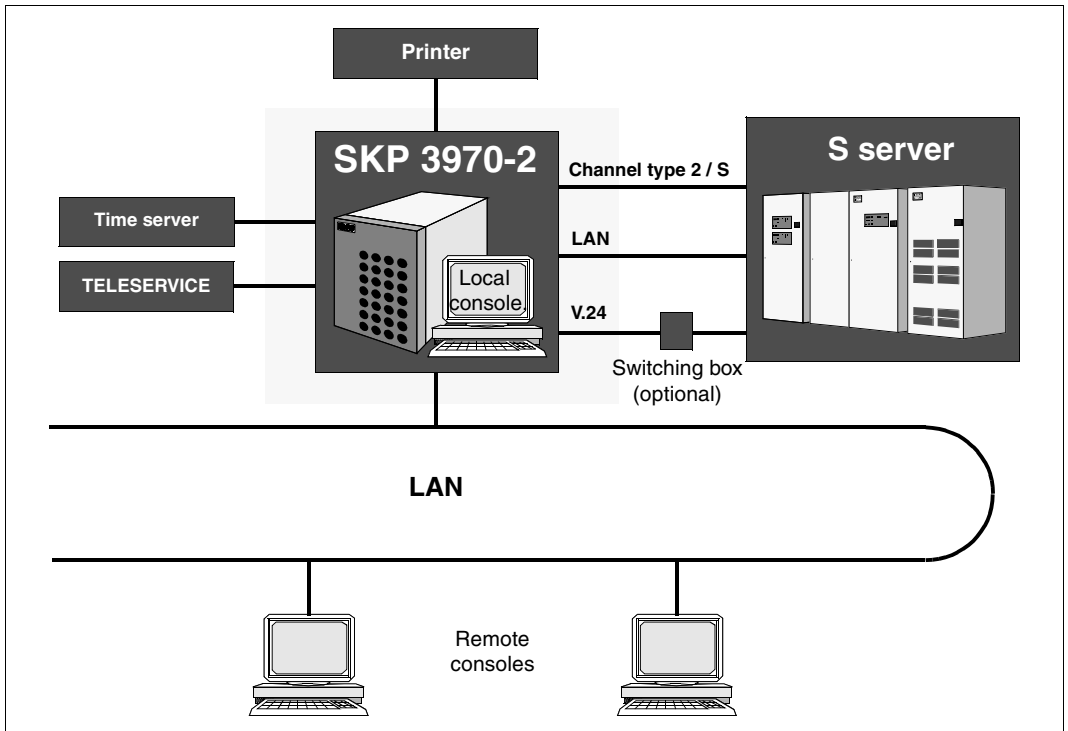


Figure 9: SCP 3970-20 (SINIX-2000) control concept with local and remote consoles (S servers)

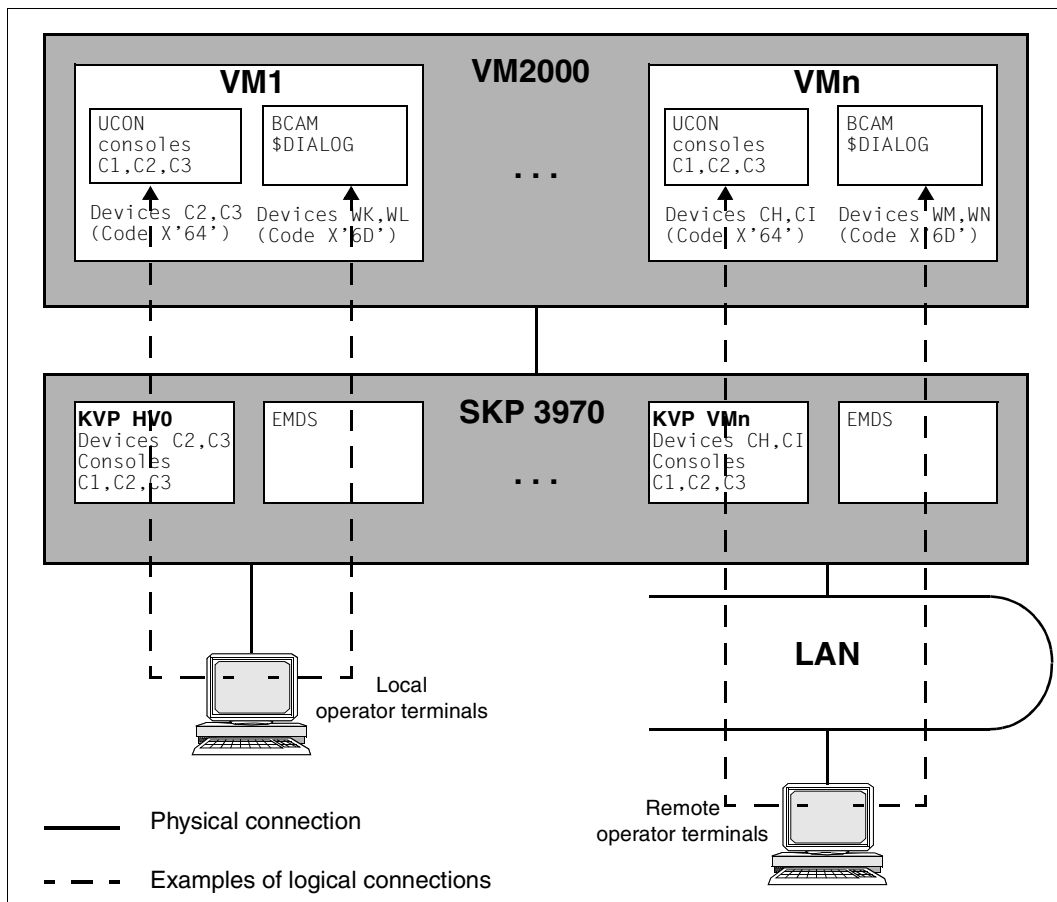


Figure 10: Example for Communication with guest systems using SCP 3970 (S servers)

For guest system operation with the SCP under VM2000, the following preparations must be made:

1. On the SKP a separate KVP with the corresponding name `VMn` is configured for each guest system which is to be operated on a `VMn` ($n=1..99$, VM index). Each KVP is made known to BS2000/OSD through its two mnemonic device names.

When a KVP is configured, user IDs are created automatically in the SKP system and mnemonic console names are assigned to these user IDs. At least the user IDs `KVP cons0` with the console `C0` and `cons1` with the console `C1` belong to the preinstalled KVP `HV0`. With all other KVPs the KVP name `VMn` is appended to the user ID, e.g. for a KVP `VM2` at least the user IDs `cons0vm2` with console `C0` and `cons1vm2` with console `C1` are created. Further user IDs with freely selectable console specifications can be created.

Information on KVP generation can be found in the relevant SCP manual [20].

2. At hardware generation of the server the device pairs that represent the KVPs set up in the SCP are generated with identical mnemonic device names (device type code 64, `SKP2`) and consecutive device addresses.

Such device pairs can also be added to the system during ongoing operation by means of a dynamic I/O configuration modification, see [page 361](#). Devices which are configured dynamically can only be used in guest systems if these are rebooted following a configuration modification. The first time a guest system is started (`/START-VM`), the two mnemonic KVP device names must be specified explicitly in the `MAIN-CONSOLE` operand.

3. The consoles of the KVP that is assigned to the VM via its device pair are made known to the guest system via its startup parameter service (`OPR` parameter `set, DEFINE-CONSOLE` statement).

The following actions are necessary in VM2000 operation:

1. Establishing the connection from the KVP to the guest system by assigning the two KVP devices to the guest system, e.g. with the VM2000 command `/ADD-VM-DEVICES`.
2. Log in on the SKP under a user ID which is configured for BS2000 console access, e.g. `cons0vm5`. In selecting the user ID you simultaneously define which mnemonic name of the console (`C0` in the example because of `cons0vm5`) and which KVP (`VM5` in the example because of `cons0vm5`) is used.

For further information, see the relevant SCP manual [20].

After logon to a console, the KVP reports this console to the BS2000 operating system of the guest system as operational. Multiple guest system operators can log in from different KVP consoles under the same user ID; in this case they then all receive the same outputs.

3. When the guest system is started (`/START-VM`), the two mnemonic device names of the KVP must be specified in the `MAIN-CONSOLE` operand, see [page 628](#).

In this way, guest systems under VM2000 can be operated from local and remote consoles of the SCP.

Dialog connections

If the data communication system is active in the guest system, dialog connections to the guest systems can be operated from the consoles of the SCP via EMDS.

Teleservice

When you use TSB2000 as of V3.0, teleservice access can be set up to any VM.

If you use TELE-X-LINK (hardware teleservice) instead of TSB2000, a shell is set up by means of which a console or EM9750 emulations can be started.

5.1.3.2 KVP consoles on SQ servers

Guest system operation with KVP consoles via X2000 on an SQ server is controlled by console distribution programs (KVPs). KVP parameters and devices are configured via the SQ manager in X2000 when the system is installed and when VM2000 is installed, see the manual “SQ Series Business Servers, Operation and Administration” [18]. The console screens are integrated into the SQ manager.

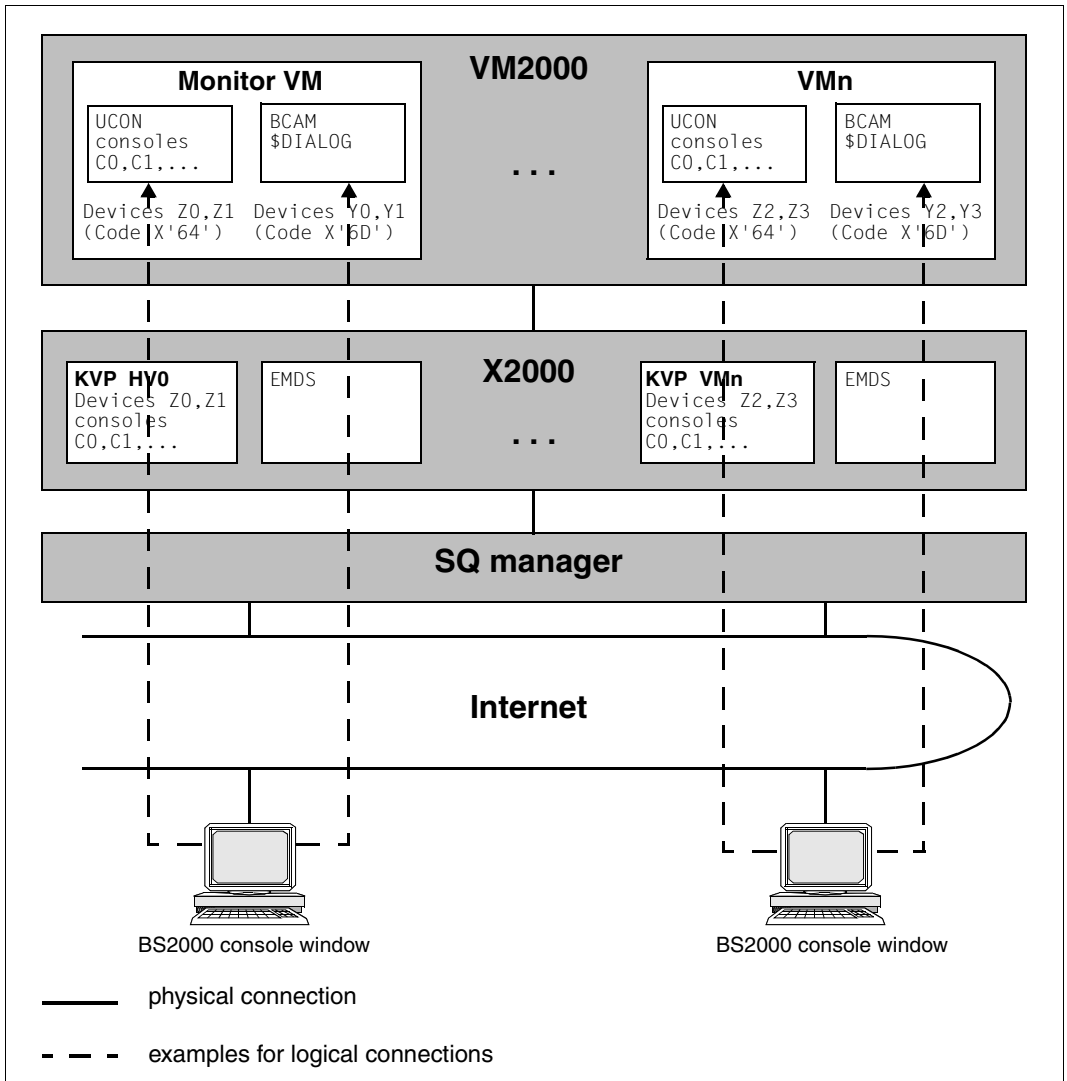


Figure 11: Example of communication with guest systems via KVP consoles (SQ servers)

Preparations

The following preparations must be made to operate the guest systems using KVP consoles:

1. A separate KVP must be created via the SQ manager for every guest system to be operated. Each of these KVPs is represented, with respect to VM2000 and the BS2000 guest system, by two devices with consecutive device addresses or by its two mnemonic device names (KVP console device pair). These devices are automatically made known to VM2000 and the BS2000 operating system.

Such device pairs can also be added to the system during ongoing operation, see [page 362](#). They can only be used in guest systems if these are rebooted following a change to the configuration. In this case, the first time a guest system is started (`/START-VM`), the two mnemonic KVP device names must be specified explicitly in the `MAIN-CONSOLE` operand.

User IDs are automatically configured in X2000 for the consoles offered by a KVP. Passwords must be defined for these KVP IDs using the SQ manager.

2. The consoles of the KVP which is assigned to the VM via its KVP console device pair must be made known to the guest system via the guest system's startup parameter service (OPR parameter set, `DEFINE-CONSOLE` statement).

Activities in VM2000 operation

The following actions are necessary in VM2000 operation:

1. Creating the connection from the KVP to the VM by assigning both KVP devices to the VM, e.g. using the VM2000 command `/ADD-VM-DEVICES`.
2. Logging an administration PC in to the SQ manager and opening a BS2000 console window of the KVP assigned to the VM.

After logging in to a console, the KVP reports to the BS2000 guest system that this console is operable. Multiple guest system operators can log in from different KVP consoles under the same user ID; in this case they then all receive the same outputs. After console mode has been accessed successfully, the current (most recent) console screen is displayed for the guest system operator.

3. When the VM is started with `/START-VM`, both mnemonic device names of the KVP device pair must if necessary be specified explicitly in the operand `MAIN-CONSOLE`, see [page 628](#).

Remote service, Teleservice

Only the service ever makes settings for Remote service / Teleservice. Service access can be set up for every VM, see the manual “SQ Series Business Servers, Operation and Administration” [[18](#)].

5.1.4 VM2000 administration application \$VMCONS

The administration application \$VMCONS handles the exchange of messages for operating the VMs (ADMIN dialog) and for operating the guest systems (VC dialog).



Recommendation

The ADMIN dialog is normally conducted from a privileged user task in the monitor system (see [page 91](#)).

The guest system is normally operated on a VM (operating) with KVP consoles or logical consoles (see [page 100](#)).

Functions of \$VMCONS

\$VMCONS fulfils the following functions:

- administration of the ADMIN dialogs conducted via \$VMCONS, i.e.
 - receiving VM administrator requests and passing them to the VM2000 monitor
 - receiving messages from the VM2000 monitor and passing them to the appropriate VM administrator
 - reading commands from VM2000 procedure files and monitoring their execution
- administration of all VC dialogs and their I/O operations

Embedding and implementation of \$VMCONS

\$VMCONS is a system application in the monitor system. \$VMCONS is loaded automatically once `BCAM ACTIVE` has been issued and runs under the system task `VM2C`. It is not possible to start or terminate \$VMCONS manually.

The \$VMCONS parameters are set in the parameter file of the VM2-MON subsystem (see [page 49](#)). Here, it is also possible to specify a password to authorize the establishment of a connection to \$VMCONS.

The \$VMCONS parameters can be modified during operation with `/MODIFY-VMCONS-PARAMETERS`, see [page 501](#).



As the new passwords only retain their validity to the end of the current session or until the command is entered again, the parameter file of the VM2-MON subsystem (see [page 49](#)) should also be updated.

\$VMCONS connects to the \$CONSOLE application as an authorized user program with a dynamic authorization name (see the “Introductory Guide to Systems Support” [2]). An operator ID (`VMCONS-OPERID`) and an operator role (`VMCONS-OPERATOR-ROLE`) must be defined for running \$VMCONS (see [page 60](#)).



To avoid undesirable reactions, do not assign any further operator roles to the operator ID, and use only the routing code “9” in the operator role.

The BS2000 command `/SET-MSG-SUPPRESSION` (see the “Introductory Guide to Systems Support” [2]) allows you to suppress messages to authorized user programs.

However, to ensure trouble-free operation of \$VMCONS, the command termination messages NBR074x must **not** be suppressed for the connection(s) between \$VMCONS and \$CONSOLE (identified by the authorization name assigned by UCON).

The connection to \$CONSOLE means that all of the commands and messages for the VM2000 ADMIN dialogs are recorded in the central logging file CONSLOG of the monitor system. This also applies to VM2000 commands and messages entered and output from a user task or at a monitor system BS2000 console.

The connection to \$VMCONS is established by means of an appropriate DCAM application. OMNIS is normally used for access via a terminal.

When BCAM is terminated in the monitor system, \$VMCONS is also terminated. If \$VMCONS terminates abnormally, it is restarted automatically (see [page 405](#)). After \$VMCONS has been restarted automatically, the interrupted connections to \$VMCONS must be established again.

Status indication in a job variable

The status of \$VMCONS is displayed in the program-monitoring job variable `$TSOS.SYSVMS.VMCONS`. The job variable can have the following values:

\$R \$VMCONS is running
\$T \$VMCONS is not (yet) available

The job variable is set up on the home pubset in the monitor system shortly before `SYSTEM READY` and is set exclusively by the operating system. It is shareable and protected against unauthorized modification. It is deleted when the monitor system is shut down.

In the event of abnormal system termination in the monitor system, the job variable is preserved with its last valid contents and is updated shortly before `SYSTEM READY` when the system is restarted.

5.1.4.1 Establishing a connection to \$VMCONS

The connection to \$VMCONS is established by means of an appropriate DCAM application. OMNIS is normally used for access via a terminal. A suitable network connection must be available to establish a connection.



Recommendation

Open a separate connection to \$VMCONS for each ADMIN and VC dialog (see [“Establishing multiple connections between OMNIS and \\$VMCONS”](#) on page 114).

Note

Note that each time a connection is set up to \$VMCONS, a connection to \$CONSOLE is also established implicitly. \$CONSOLE permits up to 192 connections. Some connections may already be occupied or reserved by other applications.

Establishing a connection to \$VMCONS via OMNIS

The connection to OMNIS is established at a terminal with the following OPNCON command:

```
O[PNCON] OMNIS [, <prname> bzw. <pp>/<rrr>][, PW=C'cccc' / X'xxxxxxxx']
```

The following OMNIS command is used to establish a connection to \$VMCONS (for a complete description of the command, see the “OMNIS/OMNIS-MENU” manual [11]):

```
O[PNCON] <pac>, TYP=DCAM, PTNNAME=$VMCONS [, PRONAME=<prname>]
          [, CPASS=<c-string 1..4>/<x-string 1..8>]
```

<pac> Freely-selectable OMNIS connection name.

PRONAME=<prname> If OMNIS and \$VMCONS are to run on different systems, the processor name of the system on which \$VMCONS is running must be entered here.

CPASS= Password for establishing a connection to \$VMCONS as specified in the parameter file of the VM2-MON subsystem or via /MODIFY-VMCONS-PARAMETERS (optional, see [page 53](#)).



OPNCON will be rejected by \$VMCONS if specified with `PROTOCOL=DSSIM`.

For VC dialogs whose connection is established with `TYP=UCON`, the OMNIS functions “color control” and “message tables” are available, see [“OMNIS functions “color control” and “message tables””](#) on page 122.

After a connection has been established successfully, \$VMCONS sends the VM2000 welcome screen as an OPNCON response (see note on the following page).

The `OPTION` command in OMNIS can be used to set whether OMNIS prefixes the connection name for outputs (<pac>:):

```
OPTION PAC={STD / NO / LINE / PREFIX}
```

The `OPTION` command in OMNIS can be used to set screen overflow control options:

```
OPTION ACK={STD / NO / YES}
```

The `CALL-KEY` function in the `OPTION` command in OMNIS can be used to set up a function for interrupting the output of messages to allow input (“K2 function”):

```
OPTION BREAK-KEY=1,CALL-INF=YES,CALL-KEY=2
```

Since the `BREAK` function for `$DIALOG` is located on the `[K2]` key by default, this function must be relocated to another key (e.g. `[K1]` as in the example above). If the newly defined `[K2]` key is pressed, message output at the OMNIS terminal is interrupted to allow input.



A connection to OMNIS can be established via the communication network or locally (using the SKP 3970 (S servers) or the SQ manager (SQ servers)).

Note on output of the VM2000 welcome screen

`$VMCONS` recognizes OMNIS on the basis of its partner characteristic, provided this characteristic starts with `OMS` (default), and outputs the VM2000 welcome screen.

You can use the OMNIS start parameter `APPLICATION PREFIX=xxx` to set a different partner characteristic for OMNIS. You can also set a different partner characteristic for a connection using the `APPNAME` operand at connection setup with `OPNCON`:

```
O[PNCON] <pac>,TYP=<typ>,PTNNAME=$VMCONS,APPNAME=pppxxxxx,
        [PRONAME=<proname>][,CPASS=<c-string 1..4>/<x-string 1..8>]
```

In this case, `ppp` (not equal to `OMS`) is the optional specification of a partner characteristic, and `xxxxx` is an optional component of `APPNAME`, which must, however, be unique for each connection. The partner characteristic of OMNIS does not begin in this case with `OMS`, and `$VMCONS` does not output a VM2000 welcome screen.

Splitting the operating levels

The ADMIN dialog and the VC dialog are independent of each other, i.e. the connections can be opened on separate terminals.

The VC dialog can also be opened if no ADMIN dialog has yet been opened.

Combining operating levels

VM2000 provides the following options for combining operating levels:

- Operation of a VM and the guest system on a VM via a connection to \$VMCONS, combining ADMIN dialog and VC dialog by issuing the VM2000 command `/BEGIN-VM-DIALOG ..., TYPE=*BOTH`
- Operation of **several** VMs and **several** guest systems at a **single** terminal via **several** connections to \$VMCONS (connections are controlled by OMNIS)
- Operation of **several** VMs and **several** guest systems via a **single** connection to \$VMCONS (multiple `/BEGIN-VM-DIALOG`)

It is recommended, however, that you establish a separate connection to \$VMCONS for each ADMIN dialog and each VC dialog (see the following section).

Establishing multiple connections between OMNIS and \$VMCONS

Since OMNIS itself is responsible for merging a number of connections on a single terminal, a separate connection between OMNIS and \$VMCONS should be established for each dialog (ADMIN dialog or VC dialog).

In this event, the OMNIS connection name `<pac>` should be chosen in such a way that it corresponds to the prefix of VM2000 outputs, i.e. `VM01, VM02, ..., VMnn` if an ADMIN dialog is to be conducted via this connection (see [page 116](#)) or `GS02, GS03, ..., GSnn` if a VC dialog is to be conducted via this connection (see [page 120](#)).

The OMNIS connection name `<pac>`: identifies the assignment of an input to a connection. OMNIS generally prefixes all output via this connection with `<pac>`.

If a number of connections to \$VMCONS are established, the appropriate connection name (`<pac>`;) must be prefixed to the input if the assignment has not been carried out for the preceding input.

Program connection to \$VMCONS via a DCAM application

In addition to terminal connection via OMNIS, a program connection to \$VMCONS is also possible via suitable DCAM applications (`PTNNAME=$VMCONS`).

DCAM applications whose `APPNAME` does not start with `OMS` do not receive the VM2000 welcome screen as a response to `OPNCON`.

After a connection to \$VMCONS has been established successfully, it is possible to open ADMIN and VC dialogs via the DCAM application.

5.1.4.2 Operating the virtual machines (ADMIN dialog via \$VMCONS)

Refer to [page 124](#) for an example of an ADMIN and VC dialog via \$VMCONS.



Recommendation

The ADMIN dialog is normally conducted from a privileged user task in the monitor system (see [page 91](#)).

The following stages are involved in operating a VM in an ADMIN dialog via \$VMCONS.

- ▶ Establish a connection to \$VMCONS via OMNIS (see [page 112](#)) or via a suitable DCAM application.
- ▶ Open the ADMIN dialog using the VM2000 command

```
<pac>:/BEGIN-VM-DIALOG VM-IDENTIFICATION=...,PASSWORD=...,TYPE=*ADMIN(...)
```

You must specify the OMNIS connection name `<pac>`: here to assign the command to the OMNIS connection. The slash (/) must always be specified to identify the VM2000 command.

If `TYPE=*BOTH` is specified, both the ADMIN and a VC dialog will be handled under the same connection name. The VM prefix is used to distinguish entries for administration of a VM and entries for administration of the guest system.

After the `/BEGIN-VM-DIALOG` command has been completed successfully, the VM2000 administrator is informed of the authorization name for this ADMIN dialog (message VMS1013). Messages for the ADMIN dialog are logged in the CONSLOG file for the monitor system under this authorization name (`@nnn, n=0..9`).

- ▶ Enter VM2000 commands to operate the VM (see [page 136](#) and [page 415](#)) or the VM2000 system as a whole (see [page 186](#) and [page 415](#)).
- ▶ Terminate the ADMIN dialog with one of the VM2000 commands

```
/END-VM-DIALOG VM-IDENTIFICATION=...,TYPE=*ADMIN,DISCONNECT=...
/DELETE-VM VM-IDENTIFICATION=...
```

When you enter `/END-VM-DIALOG`, you can specify whether the connection to \$VMCONS is to be maintained or cleared. If `TYPE=*BOTH` is specified, both the ADMIN dialog and the VC dialog are terminated.

If `/DELETE-VM` is specified, VM2000 terminates all ADMIN dialogs with the VM and all VC dialogs with the guest system on the VM.



One ADMIN dialog per VM can be opened via \$VMCONS (if there are free connections to \$CONSOLE, see *Note* on [page 112](#)). In addition, up to 15 ADMIN dialogs per VM can be opened concurrently via different, privileged user tasks.

VM prefix

The VM prefix is used to assign a command or message to a VM in the ADMIN dialog via \$VMCONS. It comprises the string `VMnn:`. Here `nn` is the VM index (2 digits, padded to the left with zeros, e.g. `VM03:`).

VM2000 adds the VM prefix to all output and messages of the ADMIN dialog via \$VMCONS. If the ADMIN dialog is opened with `TYPE=*ADMIN(OUTPUT-PREFIX=*NO)`, output of the VM prefix can be suppressed.

If a number of dialogs are conducted across a single connection to \$VMCONS, the VM2000 commands must include the appropriate VM prefix if the assignment has not already been carried out by the preceding command.

If several OMNIS connections are used, both identifiers, i.e. the OMNIS connection name and the VM prefix must be added to any input, e.g. `XY01:VM02:/SHOW-VM-ATTRIBUTES`.

It is recommended, however, that you establish a separate connection to \$VMCONS for each ADMIN and each VC dialog, see [page 114](#).

Logging

VM2000 commands, output and messages in the context of the operation of VM2000 and the VMs are logged to the monitor system's CONSLOG file (for more information on the CONSLOG file see the "Diagnostics Handbook" [4]).

The commands entered and messages output during an ADMIN dialog via \$VMCONS can be identified on the basis of the unique authorization name (`@nnn`, `n=0..9`) stored in the sender/recipient field of each entry in CONSLOG. The VM2000 administrator is informed of the authorization name at `/BEGIN-VM-DIALOG` (message `VMS1013`), at `/END-VM-DIALOG` or `/DELETE-VM` (message `VMS1012`) or if a connection is terminated abnormally (message `VMS1531`).



When an authorization name becomes free, it is immediately reassigned by UCON.

OMNIS logging provides a further type of logging (see the "OMNIS/OMNIS-MENU" manual [11]). When OMNIS logging is activated, OMNIS logs all messages it handles in the OMNIS logging file. This means that both the ADMIN dialog and the VC dialog are logged in the OMNIS logging file, but not the VM2000 messages to and from a BS2000 console in the monitor system.

For the purpose of evaluation, the OMNIS administrator must replace the logging file, make it readable with the OMNISLC utility and then make it available.

5.1.4.3 Operating the guest systems(VC dialog via \$VMCONS)

When \$VMCONS is used as a BS2000 console of the guest system, the message traffic is handled as a VC dialog via \$VMCONS (see [page 118](#)).

The guest system can be started up and operated via the VC dialog. An example of this is provided on [page 124](#).



We recommend operating the guest systems using KVP consoles or logical consoles (see [page 100](#)) instead of via \$VMCONS.

Implementing a VC dialog via \$VMCONS connection to KVP (SQ servers)

A VC dialog on SQ servers is implemented as a special connection to the KVP console of the `cons0<kvp-name>` ID of the KVP via which the guest system's IPL was initiated. BS2000 outputs to this KVP console are output via \$VMCONS. Inputs in the VC dialog are forwarded via \$VMCONS and the KVP to the guest system like inputs from this KVP console.

Implementing a VC-Dialog via virtual console (S servers)

A **virtual console** is a device that is generated as a physical console at hardware generation of the S servers but does not actually exist. If the virtual console is used as a BS2000 console of the guest system, the message traffic is executed as a VC dialog via \$VMCONS (see [page 118](#)).

Inputs/outputs via a virtual console are emulated by \$VMCONS and the VM2000 hypervisor for the relevant guest system via the physical console as message traffic.

The guest system is started up and operated via the virtual console. An example of this is provided on [page 124](#). You cannot use a virtual console for the monitor system.



We recommend operating the guest systems using KVP consoles or logical consoles (see [page 100](#)) instead of via \$VMCONS.

Generating the virtual console

Virtual consoles must be generated as devices, see [page 45](#).

Assigning the virtual console

Free devices that can be assigned to a VM as a virtual console have the suffix (VC) in the outputs of VM2000 information commands.

Virtual consoles must be assigned explicitly by means of `/ADD-VM-DEVICES` with the operand `TYPE=*VC`. They cannot be assigned implicitly. Each VM/each guest system can be assigned **one** virtual console. A virtual console cannot be assigned to the monitor VM.

Virtual consoles of a VM have the suffix (VC) in the outputs of VM2000 information commands.

Conducting a VC dialog

The following stages are involved in operating a guest system with a VC dialog:

- ▶ Establish a connection to \$VMCONS via OMNIS (see [page 112](#)) or via a suitable DCAM application.

When a connection is established to \$VMCONS with TYP=UCON, the OMNIS functions “color control” and “message tables” are also available for the VC dialog, see “[OMNIS functions “color control” and “message tables”](#)” on [page 122](#).

- ▶ Open a VC dialog with the following VM2000 command:

```
<pac>: /BEGIN-VM-DIALOG VM-IDENTIFICATION=...,PASSWORD=...,TYPE=*VC(...)
```

You must specify the OMNIS connection name <pac>: here to assign the command to the OMNIS connection. The slash (/) must always be specified to identify the VM2000 command.

Up to **eight VC dialogs** can be conducted in parallel with one guest system.

If TYPE=*BOTH is specified, both the ADMIN and a VC dialog will be handled under the same connection name. The GS prefix is used to distinguish entries for administration of the guest system and entries for administration of a VM.

- ▶ Enter BS2000 commands and statements for operating the guest system on the VM.

- ▶ Terminate the VC dialog with one of the VM2000 commands

```
VMnn: /END-VM-DIALOG VM-IDENTIFICATION=... ,TYPE=*VC,DISCONNECT=...
VMnn: /DELETE-VM VM-IDENTIFICATION=...
```

VMnn is the VM prefix for an ADMIN dialog on this connection. If VC dialogs only are conducted over this connection, the prefix VM00: must be specified to distinguish entries for the guest system.

When you enter /END-VM-DIALOG, you can specify whether the connection to \$VMCONS is to be maintained or cleared. If TYPE=*BOTH is specified, both the ADMIN dialog and the VC dialog are terminated.

When the /DELETE-VM command is issued, all ADMIN dialogs with the VM and all VC dialogs with the guest system on the VM are terminated by VM2000.



Up to **eight VC dialogs** can be conducted in parallel with one guest system. Several operating or monitoring instances can thus operate the guest system in parallel. Please observe the *Note* on [page 112](#) here.

All the VC dialogs of a guest system have the same command and message authorization. Following BCAM ACTIVE in the guest system, logical consoles (e.g. via OMNIS) with differentiated privileges and privileged user tasks can be used.

When data transfer in a VC dialog is interrupted, outputs of the guest system are recognized and are reported to the VM or VM2000 administrator in message VMS1602.

In the following cases, \$VMCONS is temporarily not available:

- automatic restart in the monitor system (see also [page 401](#))
- failure of OMNIS
- failure of the terminal or the connection to the terminal

Guest systems that are **only** operated via \$VMCONS cannot be operated during this time.

- [S] On S servers the outputs of the guest systems in this period are either buffered (if the system parameter NBMSGCSD=N is set) and output as soon as \$VMCONS is available again or only output to the CONSLOG file (NBMSGCSD=Y).
- [SQ] Outputs of the guest systems are not buffered on SQ servers during this time because the associated KVP console remains operable from the BS2000 viewpoint. In the event of a connection failure (i.e. not in the case of a restart), some messages are buffered by the VM2000 Agent. However, all outputs are stored in the CONSLOG file or in KVP logging.

GS prefix

The GS prefix is used to assign input or output to a guest system. It comprises the string `GSnn:`. Where `nn` is the index of the VM on which the guest system is running (2 digits, padded to the left with zeros, e.g. `GS03:`).

By default, VM2000 adds the GS prefix to all outputs from a VC dialog. If the VC dialog is opened with `TYPE=*VC(OUTPUT-PREFIX=*NO)`, output of the GS prefix can be suppressed.

If a number of dialogs are conducted across a **single** connection to \$VMCONS, entries made in the guest system must include the appropriate GS prefix if the assignment has not already been carried out by the preceding command. If several OMNIS connections are used, both identifiers, i.e. the OMNIS connection name and the GS prefix, must be added to any input, e.g. `XY01:GS02:P.END`

Message flow when operating the guest system (S servers)

Since all messages between the virtual console and the guest system are handled by the VM2000 hypervisor and the monitor VM (path (1) in figure 12), this places an additional burden on the VM2000 hypervisor and the monitor VM.

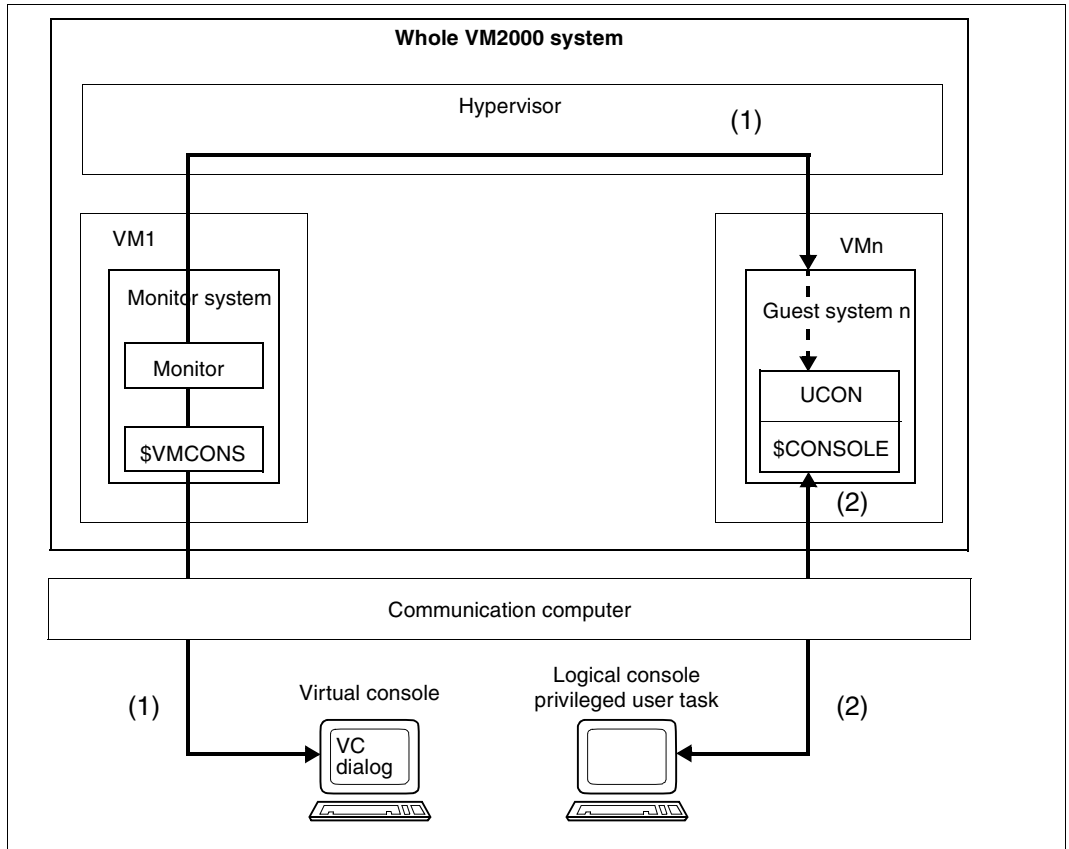


Figure 12: Message flow when operating the guest system

To relieve the burden on the hypervisor if there is heavy input/output traffic across the virtual consoles, it is possible to define logical consoles for the guest system (e.g. using OMNIS) after `BCAM ACTIVE` has been issued in the guest system (see the “OMNIS/OMNIS-MENU” manual [11]).

If the BS2000 command `/ADD-CONSOLE-FILTER FILTER=*ALL,ROUTING-CODE=*ALL` is entered in a VC dialog, the virtual console is placed in the so-called NOINF status. In this status, all unanswered messages which are to be distributed (indicated by % and to be sent via a routing code) are suppressed.

Normal operating is then carried out on the logical consoles (path (2) in [figure 12](#)). The virtual console is then only used to display emergency messages and to eliminate problems if a logical console fails.

OMNIS functions “color control” and “message tables”

The functions “color control” and “message tables” are described in detail in the “OMNIS/OMNIS-MENU” manual [[11](#)].

The connection to \$VMCONS is established, as described on [page 112](#), via the OMNIS command `OPNCON`, but with specification of `TYP=UCON`. The guest systems can then be operated like UCON partners by OMNIS with color control and message tables.



Recommendation for dialog design with the UCON partner type

When a connection is established with the partner type UCON, the OMNIS functions “color control” and “message tables” are also activated for the virtual console. Since these functions require a specific message format, you should note the following:

- Suppress the output of the VM2000 welcome screen when the connection is established by specifying a partner characteristic other than `OMS` (see the note on [page 113](#)).
- **Only conduct one VC dialog** (`/BEGIN-VM-DIALOG VM-ID=... , PASSWORD=... ,TYPE=*VC`) via this connection to \$VMCONS with the partner type UCON.
- Conduct the ADMIN dialog via another connection to \$VMCONS with the partner type DCAM or conduct the ADMIN dialog from a privileged user task.

Color control

Depending on the `DISPLAY-MODE`, the messages of the guest system are colored according to their meaning. The `DISPLAY-MODE` can be set with the operand of the same name in the OMNIS commands `SET`, `OPTION` and `DECLARE-TERMINAL`.

Message tables

Console messages of a guest system are received by OMNIS via the UCON interface and are output in the VC dialog. OMNIS message tables automate operation of the guest system. Message tables allow you to

- have receipt of a message signaled acoustically (BELL=YES)
- suppress output of unimportant messages (DISPLAY=NO)
- reply to messages automatically (REPLY='&VMP:&TSN.<text>')
- respond to messages with an operator command (REPLY='&VMP: /<cmd>')

Like the OMNIS placeholder &TSN, with which the reply is directed to the task that triggered the message, the placeholder &VMP is also available. However, this only has to be used when, contrary to the recommended procedure, **several** VC dialogs are conducted using only a **single** connection to \$VMCONS. &VMP can then be used to send the reply to the guest system that sent the message (REPLY='&VMP: . . . ').

The prerequisite for this is that these VC dialogs must be working with a GS prefix, i.e. they must have been opened with /BEGIN-VM-DIALOG . . . ,OUTPUT-PREFIX=*YES.

The message tables are normally predefined and are set up when OMNIS is started (OMNIS startup file). However, they can be created or modified during operation (OMNIS command MDEF, where the INSERT operand has no relevance). The OMNIS command INF MTAB provides information on current message tables.

5.1.4.4 Example of ADMIN and VC dialog via \$VMCONS

This example shows the operation of the whole VM2000 system and the VM2 by the VM2000 administrator (ADMIN dialog with the monitor VM via \$VMCONS) at an S server.

The guest system on the second VM is booted and started via a separate OMNIS connection in the VC dialog.



For an example of the ADMIN dialog via a privileged user task, see [page 95](#).

Tracer listing

```
@: OMS0001 USER IS CONNECTED TO OMNIS (TERMINAL=(STAT0815/##000035); TID= 10)
@: OMS0002 OMNIS VERSION <ver> READY _____ (1)
@: option logging=yes _____ (2)
@: option input-logging=send,output-logging=send
@: option pac=std _____ (3)
@: option break-key=1,call-inf=yes,call-key=2 _____ (4)
@: opncon vm01,typ=dcam,ptnname=$vmcons _____ (5)
@: OMS0012 'OPNCON' COMMAND ACCEPTED (PAC=VM01, PID= 11,
APPNAME=(OMS00011/SHIRKVM1))
@: OMS0014 'OPNCON' COMMAND EXECUTED SUCCESSFULLY
(PARTNER=($VMCONS/SHIRKVM1), PAC=VM01) _____ (6)
```

- (1) Via a terminal emulation the connection is established to OMNIS in the monitor system.
- (2) The OMNIS logging is activated.
- (3) The standard output of the OMNIS connection name (<pac>:) is chosen.
- (4) The “K2 function” is set up under OMNIS.
- (5) The connection to \$VMCONS is established via OMNIS with the connection name VM01. The password for access to \$VMCONS is not set up.
- (6) The connection was established successfully. \$VMCONS sends the welcome screen.

```

      VVVVV      VVVVV      MMMMMM      MMMMMM
      VVVVV      VVVVV      MMMMMMMMMM      MMMMMMMMMM
      VVVVV      VVVVV      MMMMMMMMMM      MMMMMMMMMM
      VVVVV      VVVVV      MMMMM      MMMMM      MMMMM      MMMMM
      VVVVV      VVVVV      MMMMM      MMMMM      MMMMM      MMMMM
      VVVVV      VVVVV      MMMMM      MMMMM      MMMMM      MMMMM
      VVVVV      VVVVV      MMMMM      MMMMM      MMMMM      MMMMM
      VVVVV      VVVVV      MMMMM      MMMMM      MMMMM      MMMMM
      VVVVV      VVVVV      MMMMM      MMMMM      MMMMM      MMMMM
      VVVVV      VVVVV      MMMMM      MMMMM      MMMMM      MMMMM
      VVVVV      VVVVV      MMMMM      MMMMM      MMMMM      MMMMM
      VVVVV      VVVVV      MMMMM      MMMMM      MMMMM      MMMMM
      VVVVV      VVVVV      MMMMM      MMMMM      MMMMM      MMMMM
      VVVVV      VVVVV      MMMMM      MMMMM      MMMMM      MMMMM
      VVVVV      VVVVV      MMMMM      MMMMM      MMMMM      MMMMM

      22222222      00000000      00000000      00000000
      2222      2222      0000      0000      0000      0000      0000      0000
      222      2222      0000      0000      0000      0000      0000      0000
      2222      0000      0000      0000      0000      0000      0000      0000
      2222      0000      0000      0000      0000      0000      0000      0000
      2222      0000      0000      0000      0000      0000      0000      0000
      2222      0000      0000      0000      0000      0000      0000      0000
      2222      222      0000      0000      0000      0000      0000      0000
      2222222222      00000000      00000000      00000000
  
```

/

COPYRIGHT (C) FUJITSU TECHNOLOGY SOLUTIONS
ALL RIGHTS RESERVED

```

vm01:/begin-vm-dialog vm-id=1,type=*admin(output-prefix=*no) _____ (7)
VM01:VMS3000 VM2000 COMMAND /BEGIN-VM-DIALOG FOR
      VIRTUAL MACHINE (1,MONITOR) PROCESSED SUCCESSFULLY
VM01:NBRO740 COMMAND COMPLETED 'BEGIN-VM-DIALOG';
      (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date> _____ (8)
  
```

- (7) The VM2000 administrator opens the ADMIN dialog with the monitor VM. The OMNIS connection name **vm01** must be specified here for assignment of the input to the OMNIS connection. The slash (/) to introduce a command must always be specified. Output of the VM prefix is suppressed since in this example a separate connection to \$VMCONS with an OMNIS connection of the same name is set up for every dialog under OMNIS.
- (8) The standardized command termination message is output.

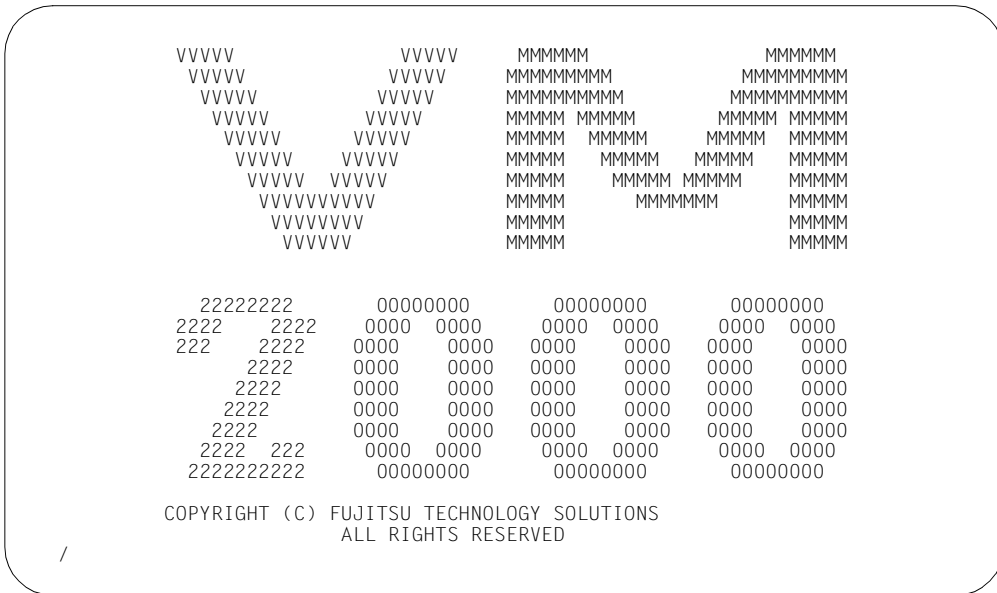
```

/CREATE-VM memory-size=128,vm-name=system2,
      cpu-quota=30,password=c' sys2' _____ (9)
VM01:VMS3050 VIRTUAL MACHINE (2,SYSTEM2) CREATED
VM01:VMS4615 VIRTUAL MACHINE (2,SYSTEM2) ASSIGNED TO CPU POOL '*STDPPOOL'
VM01:VMS3000 VM2000 COMMAND /CREATE-VM FOR
      VIRTUAL MACHINE (2,SYSTEM2) PROCESSED SUCCESSFULLY
VM01:NBR0740 COMMAND COMPLETED 'CREATE-VM';
      (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
/add-vm-devices vm-id=system2,
      units=*pubset-devices(pubset=*by-pubres-device(unit=d2)) _____ (10)
VM01:VMS4005 DEVICE 'D2' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
VM01:VMS4220 PUBSET 'VM02' WITH PUBRES DEVICE 'D2'
ASSIGNED TO VM (2,SYSTEM2)
VM01:VMS3000 VM2000 COMMAND /ADD-VM-DEVICES FOR
      VIRTUAL MACHINE (2,SYSTEM2) PROCESSED SUCCESSFULLY
VM01:NBR0740 COMMAND COMPLETED 'ADD-VM-DEVICES';
      (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
/add-vm-devices vm-id=system2,units=(z2,z3) _____ (11)
VM01:VMS4005 DEVICE 'Z2' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
VM01:VMS4005 DEVICE 'Z3' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
VM01:VMS3000 VM2000 COMMAND /ADD-VM-DEVICES FOR
      VIRTUAL MACHINE (2,SYSTEM2) PROCESSED SUCCESSFULLY
VM01:NBR0740 COMMAND COMPLETED 'ADD-VM-DEVICES';
      (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>

```

- (9) A mono-VM is initialized by the VM2000 administrator. The VM receives the VM name SYSTEM2. The password SYS2 is defined for dialog access. The OMNIS connection name need no longer be specified, as it was set with the last input.
- (10) The VM2000 administrator assigns VM2 the disk device(s) of the home pubset by specifying the pubres device. The pubset is detected by VM2000.
- (11) The VM2000 administrator assigns the KVP device pair to VM2.

```
@:opncon gs02,typ=ucon,ptnname=$vmcons _____ (12)
@: OMS0012 'OPNCON' COMMAND ACCEPTED (PAC=GS02, PID= 12,
APPNAME=(OMS00012/SHIRKVM1))
@: OMS0014 'OPNCON' COMMAND EXECUTED SUCCESSFULLY
(PARTNER=( $VMCONS/SHIRKVM1), PAC=GS02)
```



```
gs02:/begin-vm-dialog vm-id=system2,password=c'sys',
type=*vc(output-prefix=*no) _____ (13)
VM01:VMS1033 PARTNER 'OMS00012 SHIRKVM1' HAS STARTED A VC DIALOG WITH
VIRTUAL MACHINE (2,SYSTEM2) _____ (14)
VMS3000 VM2000 COMMAND /BEGIN-VM-DIALOG FOR
VIRTUAL MACHINE (2,SYSTEM2) PROCESSED SUCCESSFULLY
NBR0740 COMMAND COMPLETED 'BEGIN-VM-DIALOG';
(RERESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
```

- (12) Via OMNIS a further connection to \$VMCONS is established as TYP=UCON with the connection name GS02. This connection could also be established from a separate terminal to separate the operating levels. The prefix @ must be specified, as the statement is directed to OMNIS.
- (13) Only the VC dialog is opened. Output of the GS prefix is suppressed. The OMNIS connection name GS02 must be specified here for assignment of the input to the OMNIS connection.
- (14) The VM2000 administrator is informed that the connection has been established.

```

vm01:/start-vm ipl-unit=d2,vm-id=system2,information-byte=*dialog  (15)
VM01:VMS3000 VM2000 COMMAND /START-VM FOR
        VIRTUAL MACHINE (2,SYSTEM2) PROCESSED SUCCESSFULLY
VM01:NBR0740 COMMAND COMPLETED 'START-VM';
        (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
GS02: %P.NSI1106 *** BS2000 DIALOGUE STARTUP ***  (16)
GS02: ?P.NSI1110 ENTER OPTIONS OR EOT.
        REPLY (UNLOCK,TEST,ALLDISK,DRV-SELECT,CREATE-DRV,IPL-CONF)
gs02:p.  (17)
GS02: %P.NSI3135 IPL DISK-SETUP READ FROM IPL-CONF
        PREPARED <date> <time>
GS02: %P.NSI1143 001 PUBVOLS OF HOME-PVS VM02 WITH SYSID 70 ONLINE
GS02: %P.NSI1145 VM02.0 <date> <time> ON D2
GS02: ?P.NSI1126 CHECK IF AUTHORIZATION FOR DIALOGUE STARTUP
        SHALL BE RESTRICTED TO THIS CONSOLE CO. REPLY (Y; N; EOT=N)
p.
GS02: %P.NSI1153 STATE OF PROCESSORS ONLINE:
GS02: %P.NSI1155 CPU 00 ONLINE, ATTACHED (IPL CPU)
GS02: %P.NSI1163 LOCAL DATE = <date>, TIME = <time> FROM SVP
GS02: %P.NSI1180 LOAD ADDRESS OF SYSPRG.STRT.<ver> = 0075C000
GS02: %P.NSI00E6 FILE SYSREP.STRT.<ver> IS USED AS REP FILENAME
GS02: ?P.NSI0050 SPECIFY PARAM FILE OR DEVICE.
        REPLY ( EOT (USE STANDARD FILE);FN=FILENAME,(VOL=VSN);CONS;END )
...  (18)

```

- (15) The VM2000 administrator starts the guest system on VM2 (change of the OMNIS connection name). The pubres device of the home pubset is selected as the IPL unit. A dialog startup is to be performed. Operation of the guest system is done in the VC dialog.
- (16) Outputs of the guest system appear in the VC dialog (OMNIS connection name GS02).
- (17) Inputs to the guest system (change of the OMNIS connection name).
- (18) The guest system continues to run.


```

gs02:vm00:/end-vm-dialog vm-id=system2 _____ (19)
VM01:VMS1032 PARTNER HAS TERMINATED A VC DIALOG WITH
          VIRTUAL MACHINE (2,SYSTEM2) _____ (20)
          VMS3000 VM2000 COMMAND /END-VM-DIALOG FOR
          VIRTUAL MACHINE (2,SYSTEM2) PROCESSED SUCCESSFULLY
          NBR0740 COMMAND COMPLETED 'END-VM-DIALOG';
          (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
@: OMS0006 CONNECTION TO PARTNER ($VMCONS/SHIRKVM1) LOST
          (PAC=GS02; CODE=00000000) _____ (21)
vm01:/end-vm-dialog _____ (22)
VM01:VMS3000 VM2000 COMMAND /END-VM-DIALOG FOR
          VIRTUAL MACHINE (1,MONITOR) PROCESSED SUCCESSFULLY
VM01:NBR0740 COMMAND COMPLETED 'END-VM-DIALOG';
          (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
@: OMS0006 CONNECTION TO PARTNER ($VMCONS/SHIRKVM1) LOST
          (PAC=VM01; CODE=00000000)

```

- (19) The VC dialog is terminated. The OMNIS connection name `GS02` would only have to be specified here if the OMNIS connection were changed. Specification of `VM00` as a replacement prefix is necessary, as only a VC dialog (**TYPE=*VC**) is conducted via the OMNIS connection `GS02` and so no VM prefix is defined. If an ADMIN dialog (also) were conducted via this connection, its VM prefix would have to be specified.
- (20) The VM2000 administrator is informed that the connection has been lost.
- (21) OMNIS reports the lost connection to \$VMCONS.
- (22) The VM2000 administrator terminates the ADMIN dialog with the monitor VM.

5.1.5 Automating VM2000 and VM administration

VM2000 offers many different options for automating the administration of the virtual machines and the whole VM2000 system.

Routine tasks

Recurring tasks can be automated under VM2000 through:

- S procedures (software product SDF-P)
These extend the command language of BS2000/OSD to a programming language. If used under a user ID in the monitor system with the appropriate privilege, S procedures can contain BS2000 and VM2000 commands. With the OPERATING privilege, they can also contain BS2000 operator commands.

VM2000 supports administration via S procedures through:

- command return codes, see [section “Command return codes” on page 423](#)
- guaranteed messages, see [section “Guaranteed messages” on page 424](#)
- SHOW output in structured S variables, see [section “SHOW output in S variables” on page 425](#)

For an example of an S procedure under VM2000, see [page 133](#). In this procedure, return codes and SHOW outputs in S variables are processed.

- BS2000 procedures (non-S procedures)
These can be called by the VM2000 or VM administrator with the BS2000 command `/CALL-PROCEDURE` or `/ENTER-PROCEDURE` from a privileged user task, see [section “Operating the virtual machines \(ADMIN dialog from user task\)” on page 91](#). These BS2000 procedures contain BS2000 and VM2000 commands. If they are used under a user ID with the OPERATING privilege, they can also contain BS2000 operator commands.
- RUN files
These can be called by the VM2000 administrator with `/RUN` from a BS2000 console in the monitor system.
RUN files contain VM2000 commands and BS2000 operator commands.
- VM2000 procedures
These can be used by the VM2000 or VM administrator in the ADMIN dialog via `$VMCONS`, see `/CALL-VM-PROCEDURE` on [page 449](#).
VM2000 procedures contain VM2000 commands.

- Backup file for the VM2000 configuration
You can back up the current VM2000 configuration to a file and restore the backed-up VM2000 configuration after the overall VM2000 system has been (re)started, see [page 363](#).

Monitoring with job variables

The routine monitoring of virtual machines and main memory reconfigurations can be automated under VM2000 by job variables (MONJVs), see [page 148](#) and [page 272](#). You will find sample applications there.

Asynchronous events

The reaction to asynchronous events can be automated under VM2000. For this purpose, certain messages which are important for event-controlled automation are always output via routing code “9”.

Regardless of whether a VM2000 administrator exists (ADMIN dialog with monitor VM), the following messages are sent for the VM2000 administrator, always using **routing code “9”**.

VMS0021	VMS0024	VMS0025	VMS0026	VMS0027	VMS2011	VMS2023	VMS2033
VMS2037	VMS2050	VMS2051	VMS2052	VMS2053	VMS2100	VMS2104	VMS4065
VMS4066	VMS4075	VMS4077	VMS4094	VMS4095	VMS4342	VMS4343	VMS4344
VMS4510	VMS4511	VMS4512	VMS4514	VMS4515	VMS4551	VMS4552	VMS4553
VMS4554	VMS4555	VMS4556	VMS4557	VMS4558	VMS4559	VMS4560	VMS4561
VMS4562	VMS4563	VMS4564	VMS4620	VMS4621	VMS6002	VMS6003	VMS6020
VMS6021	VMS6023	VMS6094					

The following products can receive these messages and react automatically to the events associated with them:

- OMNIS-PROP, a special exit routine of the product OMNIS, see the “OMNIS” manual [11].

OMNIS-PROP allows you to declare times, time intervals and system messages as events. When these events occur, OMNIS-PROP activates administration procedures which perform operator tasks. To use OMNIS-PROP, you must have some knowledge of Assembler.

- PROP-XT, a software product used in data center automation, see the “PROP-XT” manual [14].

PROP-XT allows programmed operating by means of administration procedures which were created with the convenient language resources of SDF-P.

Administration procedures can thus implement via PROP-XT an event-driven VM2000 administration in parallel with the synchronous command processing in S procedures. \$VMCONS or a logical console are set up as the administration object under PROP-XT. Via this connection, the asynchronous messages are received by VM2000 and assigned an automated reaction, see the example on [page 133](#).

Up to 15 VM administrators per VM can begin the ADMIN dialog with the VM from different privileged user tasks. Several PROP-XT administration procedures can thus be used in parallel with the conventional VM or VM2000 administration for the automation of VM2000 operation.

Example

The following S procedure shows the use of PROP-XT for VM2000 administration in conjunction with a reconfiguration of the main memory. For an analogous sample procedure using MONJV, see [page 274](#).

```

/SET-PROC-OPT
/" *****"
/" ** PROCEDURE VMMEM FOR GUEST SYSTEMS WITH VM-ID=2 AND 3 *****"
/" *****"
/" ** THE PROCEDURE PERFORMS A MEMORY RECONFIGURATION. "
/" ** THE ASYNCHRONOUS TERMINATION MESSAGES ARE PROCESSED "
/" ** VIA PROP-XT. "
/" ** IMPORTANT STEPS: "
/" ** - ESTABLISH CONNECTION TO PROP-XT "
/" ** - ESTABLISH CONNECTION TO VM2000 "
/" ** - CHECK CURRENT MEMORY SIZE "
/" ** - REDUCE MEMORY OF VM2 "
/" ** - HANDLE ASYNCHRONOUS END VIA PROP-XT "
/" ** - EXTEND MEMORY OF VM3 "
/" ** - HANDLE ASYNCHRONOUS END VIA PROP-XT "
/" ** - SHUT DOWN CONNECTION TO PROP-XT "
/" ** - SHUT DOWN CONNECTION TO VM2000 "
/" *****"
/" *****"
/DECLARE-VARIABLE NAME=SYSPOP(TYPE=STRUCTURE)
/DECLARE-VARIABLE NAME=SHVAS(TYPE=STRUCTURE),MULT-E=*LIST
/DECLARE-VARIABLE NAME=MIMEMSI(TYP=INT)
/DECLARE-VARIABLE NAME=TXT(TYPE=STRING,INIT-VALUE='OK')
/DECLARE-VARIABLE NAME=CONT(TYPE=BOOLEAN,INIT-VAL=TRUE)
/ASSIGN-SYSLST LST.VMMEM
/MODIFY-JOB-OPTIONS LOG=*PAR(LIST=Y)
/BEGIN-BLOCK
/START-SUBSYSTEM PROP-XT,SYNC=Y
/IF-CMD-ERROR;END-IF
/"** LOG ON TO PROP-XT *"
/BEGIN-PROP-PROCESS VMMEM
/"** ESTABLISH CONNECTION TO LOCAL UCON *"
/START-PROP-OBJECT-MON OBJ-NAME=CONSOLE,OBJECT=*OPER-
/ (OPERATOR-ROLE=SYSADM)
/IF (SYSPOP.MAINCODE <> '0000')
/ TXT = 'ERROR IN START-P-O-M CONSOLE ** RC: &(SYSPOP.MAINCODE)'
/ GOTO LABEL=ERROR
/END-IF

```

```

/ "*** LOG EVENTS TO PROP-XT "
/ "*** VM2000 REPORTS EXECUTION OF THE COMMAND TO THE CONSOLE ASYNCHRONOUSLY "
/ "*** WITH VMS4094 (POSITIVE ACKNOWLEDGMENT) OR VMS4095 (NEGATIVE
ACKNOWLEDGMENT) "
/ "*** . THEREFORE WAIT EXPLICITLY FOR THESE TWO MESSAGES "
/ "*** (THEREFORE: LOG ON AS EVENTS!). "
/START-PROP-EVENT-MON EV-NAME=VMS4094,SELECT-EV=*FROM-OBJ(-
/ OBJ-NAME=CONSOLE,EVENT-DATA=*SYSTEM-MSG(MSG-ID=VMS4094))
/START-PROP-EVENT-MON EV-NAME=VMS4095,SELECT-EV=*FROM-OBJ(-
/ OBJ-NAME=CONSOLE,EVENT-DATA=*SYSTEM-MSG(MSG-ID=VMS4095))
/BEGIN-VM-DIALOG VM-ID=1 "ESTABLISH CONNECTION TO VM2000 "
/ "*** CHECK CURRENT MEMORY SIZE "
/EXEC-CMD (SHOW-VM-ATTR VM-ID=2,INF=*MEMORY),STRUCT-OUT=SHVAS
/IF (SHVAS#.MIN-MEM-SIZE = '*UNDEF')
/ TXT = 'MEMORY CONFIGURATION CURRENTLY NOT POSSIBLE FOR VM-ID=2'
/ GOTO LABEL=ERROR
/END-IF
/MIMEMSI=INT(SHVAS#.MIN-MEM-SIZE)
/IF (SHVAS#.MEM-SIZE = MIMEMSI)
/ TXT = 'CURRENT MEMORY SIZE IS MINIMUM MEMORY SIZE'
/ GOTO LABEL=ERROR
/END-IF
/ "*** CHECK MEMORY SIZE OF VM-ID=3 "
/EXEC-CMD (SHOW-VM-ATTR VM-ID=3,INF=*MEMORY),STRUCT-OUT=SHVAS
/IF (SHVAS#.MIN-MEM-SIZE = '*UNDEF')
/ TXT = 'MEMORY CONFIGURATION CURRENTLY NOT POSSIBLE FOR VM-ID=3'
/ GOTO LABEL=ERROR
/END-IF
/ "*** REDUCE MEMORY OF VM-ID=2 "
/REDUCE-VM-MEMORY VM-ID=2,NEW-MEM-SIZE=*MINIMUM
/IF-CMD-ERROR
/ TXT = 'ERROR IN REDUCE-VM VM-ID=2'
/ GOTO LABEL=ERROR
/END-IF
/ "*** SYSTEM MUST WAIT FOR VMS4094 OR VMS4095"
/WHILE (CONT=TRUE)
/ WAIT-F-PROP-EV EV-NAME=(VMS4094,VMS4095),TIME-LIM=300
/ IF (SYSPOP.MAINCODE <> '0000')
/ TXT = 'ERROR IN WAIT-PROP-EVENT (REDUCE-VM-MEMORY VM-ID=2)'
/ SHOW-VARIABLE SYSPOP
/ GOTO LABEL=ERROR
/ END-IF
/ IF ((SYSPOP.EVENT-NAME = 'VMS4094') AND (SYSPOP.IO='2'))
/ CONT=FALSE
/ ELSE-IF ((SYSPOP.EVENT-NAME = 'VMS4095') AND (SYSPOP.IO = '2'))
/ TXT = 'VMS4095 IN REDUCE-VM-MEMORY VM-ID=2'
/ GOTO LABEL=ERROR
/ END-IF

```

```

/END-WHILE
/ "*** EXTEND MEMORY OF VM3 *****"
/EXTEND-VM-MEMORY VM-ID=3,NEW-MEM-SIZE=*MAXIMUM
/CONT = TRUE
/WHILE (CONT=TRUE)
/ WAIT-F-PROP-EV EV-NAME=(VMS4094,VMS4095),TIME-LIM=300
/ IF (SYSPOP.MAINCODE <> '0000')
/   TXT = 'ERROR IN WAIT-PROP-EVENT (EXTEND-VM-MEMORY VM-ID=3)'
/   SHOW-VARIABLE SYSPOP
/   GOTO LABEL=ERROR
/ END-IF
/ IF ((SYSPOP.EVENT-NAME = 'VMS4094') AND (SYSPOP.IO='3'))
/   CONT=FALSE
/ ELSE-IF (SYSPOP.EVENT-NAME = 'VMS4095')
/   TXT = 'VMSVMS4095 IN EXTEND-VM-MEMORY VM-ID=3'
/   GOTO LABEL=ERROR
/ END-IF
/END-WHILE
/STOP-PROP-EV-MON (VMS4094,VMS4095)
/IF-CMD-ERROR; END-IF
/STOP-PROP-OBJ-MON CONSOLE      "SHUT DOWN CONNECTION TO UCON"
/IF-CMD-ERROR; END-IF
/ "*** NORMALES ENDE"
/END-BLOCK
/IF-BLOCK-ERROR      "HANDLE OTHER ERRORS"
/ TXT = 'OTHER ERRORS'
/ GOTO LABEL = ERROR
/END-IF
/ERROR:
/IF (TXT <> 'OK')
/ SEND-MSG TO=OPER,MSG='*** ERROR IN VMMEM: &TXT ***'
/END-IF
/END-PROP-PROCESS      "SHUT DOWN CONNECTION TO PROP-XT"
/IF-CMD-ERROR
/END-IF
/END-VM-DIALOG VM-ID=1      "SHUT DOWN CONNECTION TO VM2000"
/IF-CMD-ERROR
/END-IF
/MOD-JOB-OPTIONS LOG=*PAR(LIST=NO)
/ASSIGN-SYSLST *PRIMARY
/EXIT-PROC

```

5.2 Operating a VM

The VM administrator operates a VM in an ADMIN dialog using VM2000 commands. A [Grouping of VM2000 commands according to function](#) is provided on [page 419](#). VM2000 commands are logged, see [page 93](#) and [page 116](#).

A VM administrator for the monitor VM is also the VM2000 administrator. He/she operates and administers the VM2000 system as a whole. He/she can also operate all the other VMs.

Status diagram of a VM

A VM can assume the following **statuses**:

NOT INITIALIZED The VM is not initialized.

INIT-ONLY The VM is initialized. Resources can now be assigned to the VM, but no guest system has been started yet (after /CREATE-VM or /RESET-VM)

RUNNING The guest system on the VM has been started and is running

IN HOLD (WAIT) The VM is in the wait state, the guest system is idle

DOWN The guest system on the VM has been terminated (after /SHUTDOWN-VM resp. shutdown or SETS in the guest system)

MIGRATING A live migration (SQ servers) is running for the VM, see [page 379](#). This status cannot be changed using VM2000 commands.

These statuses (with the exception of MIGRATING) are brought about by the following VM2000 commands or events. The commands are described in detail in the following sections.

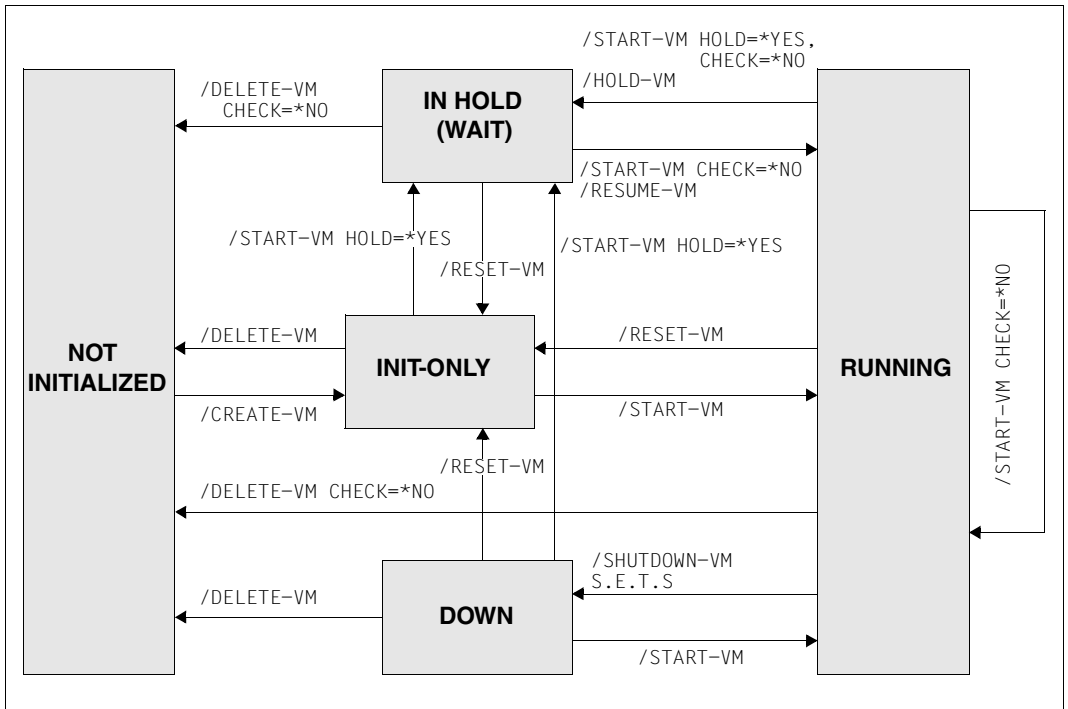


Figure 13: Status diagram of a VM


5.2.1 Initializing a VM


Before a VM can be used, it must first be initialized in VM2000. This first stage of operation is known as **initializing** a VM. A VM is initialized by means of `/CREATE-VM`, which may only be input by the VM2000 administrator. The VM administrator is allowed to terminate but not to initialize a VM.

During the initialization process, the following attributes and resources are assigned to the VM:

- VM index and VM name (identification of the VM)
- main memory size of the VM
- minimum and maximum main memory size of the VM for main memory reconfiguration
- location of the VM in the main memory of VM2000
- CPU quota and maximum CPU utilization of the VM
- Maximum IO utilization of the VM
- adding of the VM to a VM group
- assignment of the VM to a CPU pool
- multiprocessor level of the VM
- password for dialog access
- Range of commands available to VM2000 and VM administrators
- Privileges of the VM

After the `/CREATE-VM` command has been executed successfully, the VM has the status `INIT-ONLY`.

 On SQ servers a VM's firmware component is started when the VM is initialized. Despite the `INIT-ONLY` status, the VM already utilizes a minimal CPU capacity.

 The monitor VM is initialized automatically. Its attributes and resources are configured when VM2000 is installed (see [chapter "Installing VM2000" on page 43](#)).

The maximum number of VMs which can be created is dependent on the hardware, see [page 457](#). It is also displayed when `/SHOW-VM-RESOURCES INFORMATION=*CONFIGURATION` is used.

Identification of the VM

In VM2000 commands, a VM is identified by its **VM-ID**. The VM-ID can be the **VM index** or the **VM name**. VM index and VM name are assigned to a VM during initialization. VM index and VM name uniquely identify the VM.

The VM index is an integer n from 1 to 99 (hardware dependent) and identifies the VM (VM1 to VM n). The VM2000 administrator can predefine the VM index explicitly. If no VM index is specified (default), VM2000 selects the next free index. The VM index is used to manage a VM internally within VM2000.

The VM name is assigned explicitly by the VM2000 administrator (default). If no VM name is specified, VM2000 assigns the standard name VM00 nn , where nn is the VM index ($nn=01..99$). `/CREATE-VM` is rejected if a specified VM name corresponds to the standard name of another VM (e.g. `VM-INDEX=5`, `VM-NAME=VM0002`) or has already been assigned. The VM name should reflect the user or the usage mode of the VM.

Recommendations for the definition and use of VM names

The VM administrator should use the **VM name** as the VM-ID in procedures. The VM index should be avoided in procedures, as it may change in every session.

The VM name should be unique within a VM2000 installation. The VM name should not be the same as a valid user ID of the monitor system.

SQ The default name (and also the only possible VM name) of the monitor VM on SQ servers is `MONITOR`.

When a VM is configured on SQ servers, the VM name is used as the domain name. The characters #, \$ and @ should therefore no longer be used in the VM name; in the domain name they are replaced by n, s and a.

The VM name may not be the same as the domain name of an existing BS2000 VM or XenVM. In an HA cluster this condition for the domain names must apply in the entire HA cluster (see [page 376](#)).

In an HA cluster no default names should be used as VM names (exception: `MONITOR` for the monitor VM).

The name ranges for VMs, VM groups and CPU pools should be disjunctive.

If the user of the VM changes (without the VM being terminated and initialized anew), generation of accounting records can be initiated by issuing `/MODIFY-VM-ATTRIBUTES` and specifying the previous VM name. In such a case VM2000 writes BS2000 accounting records for the relevant VM and for devices assigned to it (see [page 381](#)).

Size of VM main memory

This attribute determines the main memory size for the VM (see [section “Managing main memory” on page 265](#)). The maximum main memory size under VM2000 is 1 Tbyte (terabyte; 1 Tbyte = 1024 Gbytes = 1 048 576 Mbytes).

S On S servers a main memory area begins on a 1-Mbyte boundary and its size is a multiple of 1 Mbyte.

SQ On SQ servers the size of a main memory area is a multiple of 2 Mbytes.

In addition to the main memory for a BS2000 guest system, a small amount of a VM's main memory is required for the firmware component. The main memory of a VM on SQ servers should therefore be at least 512 Mbytes in size.

Minimum size of VM main memory

The minimum main memory size should only be specified for a VM if the size of the main memory of the VM (mostly a VM which is not so important) is to be reduced while the guest system is active (see [page 270](#)).

The minimum main memory size can be increased with `/EXTEND-VM-MEMORY`. On S servers it can be decreased (implicitly) with `/REDUCE-VM-MEMORY` (see [page 507](#)).



Note on dimensioning the minimum main memory size

The minimum size of the main memory selected for a VM must be at least large enough to permit the resident memory requirements in the guest system to be satisfied. The resident memory requirement depends on whether the software product DAB is used.

The current utilization of resident memory can be determined by openSM2 from the values of the MEMORY report group (see the “openSM2” manual [8]):

`Resident Memory = TOTAL - Pageable Frames .`

SQ On SQ servers the minimum size of a VM's main memory must be at least 512 Mbytes, see above.

If the main memory of a VM is reduced to the minimum size, the load on the guest system must be reduced accordingly.

Maximum size of the VM's main memory (SQ servers)

The maximum size of the main memory should be defined for a VM only when the main memory of the VM (generally a particularly important VM) is to be extended while the guest system is active (see [page 270](#)).

If the VM's main memory is not to be extended during ongoing operation, the same value for the main memory should be selected for the maximum size (`MAX-MEMORY-SIZE`) as for the VM's main memory (`MEMORY-SIZE`).

The default value of the maximum size of the main memory is twice the size of the main memory for the VM concerned which is specified by `MEMORY-SIZE`. The maximum size of the VM's main memory is limited by the main memory which is available (output line `TOTAL REAL MEMORY SIZE` in `/SHOW-VM-RESOURCES INFORMATION=*CONFIGURATION`).

A VM can (without a message being issued) also be assigned a smaller main memory size than requested when:

- the requested value (specified explicitly or implicitly by means of the default value) is greater than the main memory which is available
- the minimum size of the VM is too small for the implicit default value (double the size of the VM's main memory)

However, when the value of the maximum size in the latter case is specified explicitly, such a memory combination is rejected (`VMS4093`).

- [S] On S servers the maximum size of a VM's main memory is ignored. A VM can always be extended to the start of the next VM or to the end of the main memory.

Location of the VM in VM2000 main memory

- [S] On S servers this attribute determines the location of the VM in the main memory of VM2000 (see [page 265](#)). The address must be a multiple of 1 Mbyte. If the position is not specified, VM2000 selects a suitable area. The location of the VM in main memory can subsequently be modified by means of `/MOVE-VM`.
- [SQ] On SQ servers the location of a VM need not be changed. Thus only the default value can be specified for this attribute (`*ANY`, the location of the VM in the main memory is not predefined).

CPU quota and maximum CPU utilization of the VM

These parameters determine the longterm distribution of the available CPU capacity on the VMs.

- [S] On S servers the CPU quota determines for a VM which does not belong to a VM group, the VM's share of the CPU capacity of the CPU pool in comparison to the VM groups and the other VMs which do not belong to a VM group. In the case of a VM which belongs to a VM group, the member CPU quota determines this VM's share of the CPU capacity of the CPU pool in comparison with the VMs of the same VM group. The CPU share of a VM can be restricted by the maximum CPU utilization of the VM or VM group.
- [SQ] On SQ servers the VM's CPU quota determines the VM's share of the CPU capacity of the CPU pool in comparison to the other VMs. The CPU share of a VM can be restricted by the maximum CPU utilization of the VM.

Further details can be found in the [section "Planning distribution of the CPU capacity to the VMs" on page 238](#).

The CPU quota and maximum CPU utilization can be modified by means of `/MODIFY-VM-ATTRIBUTES`.

Adding the VM to a VM group

A VM can be operated as a VM that does not belong to a VM group or as a member of a VM group.

- [S] On S servers the VM can be added to a VM group when it is created using the `CPU-QUOTA=*BY-VM-GROUP(. . .)` operand. In this event it is assigned a member CPU quota.
- [SQ] VM groups are not available on SQ servers.

Maximum IO utilization of the VM

The IO utilization of a VM can be limited by the maximum IO utilization of the VM.

- [S] On S servers the BS2000 subsystem IORM monitors the maximum IO utilization in the IOLVM function, see [page 360](#).
- [SQ] On SQ servers only the default value (100, unlimited utilization) can be used for this attribute.

The maximum IO utilization can be changed using `/MODIFY-VM-ATTRIBUTES`.

Assigning the VM to a CPU pool

Every VM is always assigned to precisely one CPU pool.

If the VM does not belong to a VM group, the CPU pool can be freely selected. By default (CPU-POOL-NAME=*STD operand) the VM is assigned to the standard CPU pool when it is created. The assignment of the VM to a CPU pool can be changed using /ASSIGN-VM-TO-CPU-POOL.

If the VM is added to a VM group, (S servers, CPU-QUOTA=*BY-VM-GROUP(. . .) operand), it is automatically assigned to the VM group's CPU pool (CPU-POOL-NAME=*STD operand). The assignment of the VM group to a CPU pool can be changed using /ASSIGN-VM-GROUP-TO-CPU-POOL.

Further information on CPU pools is provided in the [section “Managing CPU pools” on page 225](#).

multiprocessor level of the VM

This attribute defines the number of CPUs on which a VM is to run simultaneously. The following multiprocessor levels are supported by VM2000 (implementation limit) :

1	(MONO)	one processor (virtual CPU 0)
2	(BI)	two processors (virtual CPUs 0 and 1)
3	(TRIPLE)	three processors (virtual CPUs 0, 1 and 2)
4	(QUADRO)	four processors (virtual CPUs 0 through 3)
5	(5-Way)	five processors (virtual CPUs 0 through 4)
6	(6-Way)	six processors (virtual CPUs 0 through 5)
7	(7-Way)	seven processors (virtual CPUs 0 through 6)
8	(OCTO)	eight processors (virtual CPUs 0 through 7)
9	(9-Way)	nine processors (virtual CPUs 0 through 8, SQ servers)
.
32	(32-Way)	32 processors (virtual CPUs 0 through 31, SQ servers)

[S] On S servers the maximum multiprocessor level is 8.

The multiprocessor level of a VM must be less than or equal to the number of real normal CPUs which can be available for VM2000 operation.

Exception: see the note on the PROCESSOR=*EXTRA-AND-NORMAL operand on [page 224](#).

The virtual CPUs of the VMs that result from this are started up on the available real CPUs, see [section “Scheduling procedures” on page 249](#).

The multiprocessor level of a VM constitutes the upper limit for the maximum CPU utilization, see [page 142](#). For example, a biprocessor-VM can accommodate the CPU capacity of up to two real CPUs.

For the monitor VM, the multiprocessor level is set when VM2000 is installed (see [chapter “Installing VM2000” on page 43](#)).

Once a VM has been created, its multiprocessor level can no longer be modified.

Password for administration and operation

This attribute defines a password which must be specified by the VM administrator (in the ADMIN dialog) and the guest system operator (in the VC dialog) when a dialog is opened with `/BEGIN-VM-DIALOG`. If no password is specified, no password is required for `/BEGIN-VM-DIALOG`. The password can subsequently be modified by means of `/MODIFY-VM-ATTRIBUTES`. There are other protective features for operation using a BS2000 console.

Range of commands available to VM2000 and VM administrators

This attribute defines the range of commands available to VM2000 and VM administrators. The range of commands can be restricted (for VM2000 administrators) or extended (for VM administrators); see [page 192](#).

The range of commands can subsequently be modified by means of `/MODIFY-VM-ATTRIBUTES`.

Privileges of the VM

IO RESET privilege

The IO RESET operation is an extreme measure to overcome problems in the input/output configuration. To do this, the VM must be assigned the privilege `IO-RESET=*YES` (`/CREATE-VM` and `/MODIFY-VM-ATTRIBUTES`).

- [S] On S servers it is recommended that a VM be set up without a privilege (i.e. with `IO-RESET=*NO`) and that the privilege should only be assigned using `/MODIFY-VM-ATTRIBUTES` when required.
- [SQ] On SQ servers only the default value (`*NO`, no problem correction with IO-RESET) can be used for this attribute.

For a VM with `IO-RESET=*YES`, VM2000 takes the following actions on S servers:

1. When `/START-VM` is issued (or when the guest system is restarted), a system reset is carried out, similarly to a firmware IPL. In this case, all channels of this VM are reset in the hardware where at least one disk is assigned to the VM (either `EXCL` (exclusive) or `SH(D)` (SHARED, direct I/O)).
2. When resetting a channel via the guest system on the VM (e.g. in the case of local channel reconfiguration), the channel is reset in the hardware.
3. If `/REMOVE-VM-DEVICES` is specified with `FORCE=*YES` (called explicitly or executed during `/DELETE-VM`) all channels to which the device is connected are reset in the hardware as necessary.

For a VM with `IO-RESET=*NO`, resetting channels is emulated by the VM2000 hypervisor for the VM, and no action is taken in the hardware.

Effects on other VMs:

In the three cases listed above, all executing input/output tasks of **other** guest systems on these channels are terminated. Further execution of this guest system depends on the relevant error recovery routine in the guest system.

IO-RESET for the monitor VM:

Measure 2 is always carried out for the monitor VM.

Measure 1 is carried out when the monitor system is restarted if `IO-RESET=*YES` was specified for the monitor VM with `/CREATE-VM` (on initializing the monitor VM) or `/MODIFY-VM-ATTRIBUTES`.

IO PRIORITY privilege

Under VM2000, a VM that goes into the wait state (IDLE) after an input/output has started, for example, transfers the real CPU to another VM that is ready for operation. The input/output that has been started can be terminated in the case of fast cache media, for example, before the VM starts running again on a real CPU. The VM waits until it starts running again on a real CPU as a result of scheduling (see [page 249](#)). It can then process the result of the input/output.

A VM or guest system that is slowed down in this way can counter this effect by means of the `IO-PRIORITY=*YES` privilege.

On S servers a VM in the wait state with this privilege is put into operation again on a real CPU immediately on completion of the pending input/output. The guest system can then immediately process the result of the input/output.

The `IO-PRIORITY=*YES` privilege can be assigned to the VM at setup (`/CREATE-VM`) or later (`/MODIFY-VM-ATTRIBUTES`). It applies to all virtual CPUs of the VM.

- S On S servers it is recommended that a VM be set up without a privilege (i.e. with `IO-PRIORITY=*NO`) and that the privilege should only be assigned using `/MODIFY-VM-ATTRIBUTES` when required.
- SQ On SQ servers only the default value (`*NO`, no IO prioritization) can be used for this privilege.

The sum of the virtual CPUs of all VMs with the `IO-PRIORITY=*YES` privilege must not be greater than the number of real normal CPUs of the server.

AUTO-SNAP-ASSIGNMENT privilege

This privilege permits the guest system on a VM to assign itself snap units of a Snapshot implicitly without the VM and device being assigned the `ASSIGN-BY-GUEST` privilege or attribute.

ASSIGN-BY-GUEST privilege


This privilege determines whether the operating can implicitly assign devices of particular assignment sets to the own VM himself/herself (e.g. with `/ATTACH-DEVICE`), see [page 288](#).

To permit implicit device assignment, the VM must have the `ASSIGN-BY-GUEST` privilege for the required assignment sets (either at setup with `/CREATE-VM` or later with `/MODIFY-VM-ATTRIBUTES`).

Each device that is to be implicitly assigned must also have the attribute `ASSIGN-BY-GUEST` (`/MODIFY-VM-DEVICE-ATTRIBUTES`, see [page 491](#)). The device is also assigned to the required assignment set here.

Settings for control over the real CPU

On S servers this attribute determines whether, in the event of fixed CPU assignment (dedicated CPUs), a VM still retains control over a real CPU if the VM's virtual CPU which runs on this is inactive (interruptible wait state).


 On SQ servers only the default value (*NO) can be used for this attribute.

When `VM-ACTIVE-IDLE=*NO` (on S servers) the VM2000 hypervisor deletes the real CPU assigned if the VM's virtual CPU which runs on this is inactive (interruptible wait state IDLE).

When `VM-ACTIVE-IDLE=*AT-DEDICATED-CPUS`, the VM retains control over the real CPU assigned even if the VM's virtual CPU which runs on this is inactive (interruptible wait state IDLE).

In this case additional performance is achieved because no change of context takes place. However, this idle time is then indicated in the VM2000 accounting records, with `/SHOW-VM-STATUS` (`VM-ACTIVE` output column) and in the VM2000 report of openSM2 as a time in which the VM actively uses the real CPU.

This setting provides no additional performance if a large number of I/Os are to be expected for shared disks or on virtual consoles.

 With fixed CPU assignment, `VM-ACTIVE-IDLE=*AT-DEDICATED-CPUS` is only effective if the VM's maximum CPU utilization (see [page 142](#)) is not restricted.

The setting can also be modified later using `/MODIFY-VM-ATTRIBUTES`.

5.2.2 Monitoring a VM with job variables (MONJV)

Job variables (JVs) are memory areas used for exchanging information. They can be addressed on a command level and on a program level. They have a path name and are stored in the file catalog.

Under VM2000, job variables are used to transfer information from VM2000 to an instance which is to monitor a VM. Virtual machines can thus be monitored using job variables.

When the VM is initialized by the VM2000 administrator with `/CREATE-VM`, a monitoring job variable (MONJV) is defined. The location and validity of the MONJV correspond to the runtime environment of the sender of the command. VM2000 manages and updates the MONJV as long as the VM exists. By using appropriate procedures, the VM2000 administration can automatically react to the information stored in the MONJV and thus facilitate VM2000 administration.

When the VM has been terminated, the MONJV is retained as a user job variable.



The monitor VM cannot be monitored with a MONJV.

The software product JV, which is contained in the software package OSD/XC for SQ servers, is required for working with job variables. This is contained in the software package OSD/XC. For more information on job variables, see the manual “Job Variables” [16], especially the section “Monitoring jobs/programs with job variables”.

For an example of how to use a MONJV to monitor a VM, see [page 152](#).

MONJVs can also be used to monitor the main memory reconfiguration, see [page 272](#).

Beginning and end of monitoring

The MONJV used to monitor the VM is configured when the VM is initialized with `/CREATE-VM` by specifying the operand `MONJV`. The monitoring of the VM begins at this point; the MONJV is supplied with data by VM2000.

If a MONJV already exists with the specified name, a password can be specified in the operand `JV-PASSWORD` to check for access authorization.

The monitoring ends when the VM is terminated: i.e. with `/DELETE-VM` for the VM or when terminating VM2000 (implicit `/DELETE-VM VM-ID=*ALL`). Even when the monitoring has finished, the MONJV is retained as a user job variable without MONJV protection.

Protecting the MONJV

The operating system protects the first 128 bytes (system area) of the MONJV from unauthorized write access.

The MONJV itself is protected against deletion of or changes to the protection attributes. To prevent inconsistencies, the BS2000 systems support staff must **not** cancel this protection. If necessary (e.g. if the system has been terminated abnormally in the monitor system without a restart), the MONJV protection can be canceled with:

```
/MODIFY-JV-ATTRIBUTES JV-NAME=<monjv name>,
                    PROTECTION=*PARAMETERS(MONJV-PROTECTION=*NO)
```

When the VM has been terminated normally, the MONJV is retained as a user job variable (status \$T_) without MONJV protection.

Location of the MONJV

The MONJV can be created on any pubset in the monitor system. Valid catalog and user IDs of the MONJV depend on the runtime environment of the sender of the command.

Command entry from user task

Privileged users (TSOS) can create the MONJV under any user ID they wish. Non-privileged users can only create the MONJV under their own user ID.

If catalog and user IDs are not specified, the MONJV of VM2000 will be shared under the user ID of the sender of the command and created without a password, if no job variable with the specified name already exists there.

Command entry from KVP console or logical console

The sender of the command without an operator logon can only create the MONJV under TSOS.

If catalog and user IDs are not specified, the MONJV of VM2000 will be shared on the home pubset of the monitor system under TSOS and created without a password, if no job variable with the specified name already exists there.

For senders of commands with operator logons, the location is defined, depending on the settings of the operator logon, in the same way as for *Command entry from user task*.

Command entry via \$VMCONS

The user ID and operator role for a command entry via \$VMCONS are: VMCONS-OPERID and VMCONS-OPERATOR-ROLE (see [page 49](#)).

The location of the MONJV is defined, depending on the settings of VMCONS-OPERID and VMCONS-OPERATOR-ROLE, in the same way as for *Command entry from user task*.

*Recommendation*

When preparing to use a MONJV, the VM2000 administration must carry out the relevant administrative actions needed to coordinate the creation of MONJVs and access to the MONJVs.

If, for instance, the same MONJV is accessed by different IDs, it is advisable to have previously configured and made this MONJV known centrally with the appropriate protection attributes. The MONJV must then be addressed in the VM2000 commands and when accessing its contents, by specifying its catalog and user IDs.

Values for the MONJV

A MONJV for a VM comprises a system part (bytes 1-128) and a user part (bytes 129-256). Specific values are entered by VM2000 in the system part of the MONJV at specific times as long as the VM exists.

Byte	Meaning / possible values
1-3	Monitoring status: <ul style="list-style-type: none"> – \$S_L : /CREATE-VM for VM (Creation Started) – \$A_L : /CREATE-VM for VM terminated with error (Creation Aborted) – \$I_L : VM initialized (VM status INIT-ONLY) – \$R_L : VM running (VM status RUNNING) – \$D_L : guest system terminated on the VM (VM status DOWN) – \$H_L : VM stopped (VM status IN HOLD (WAIT)) – \$T_L : VM terminated (Terminated, VM status NOT INITIALIZED)
4	Reserved, has the value zero (0)
5-8	TSN of the job, contains spaces
9-12	Catalog ID of the monitor system home pubset (left-aligned, padded with spaces)
13-16	Reserved, contains spaces.
17	Type of MONJV: character V (VM2000)
18-20	Current system sequence number (<integer 1..999>)
21-36	Beginning of monitoring (opening the MONJV) in UTC time (format: yyyy-mm-ddhhmmss)
37-70	Reserved
71-78	Name of VM (<name 1..8>, not for status \$S and \$A)
79-81	Index of VM printable (<name 3..3>, e.g. 002)

Structure of monitor job variables for a VM

Byte	Meaning / possible values
82-86	Status of guest system: <ul style="list-style-type: none"> – NONE : guest system not loaded (for \$I and \$T) – START : guest system in startup phase (for \$R and \$H) – READY : guest system has SYSTEM READY (for \$R and \$H) – RSTRT : restart in guest system (for \$R and \$H) – NTERM : guest system terminated with SHUTDOWN (for \$D and \$H) – ATERM : guest system terminated by SETS or VM administrator (for \$D and \$H)
87-128	Reserved
129-256	Reserved for users

Structure of monitor job variables for a VM

Displaying the MONJV

You can query the contents and attributes of the MONJV using the commands and macros for job variables, e.g. /SHOW-JV and /SHOW-JV-ATTRIBUTES (see the “Job Variables” manual [16]).

The name of the MONJV is displayed in the VM2000 commands /SHOW-VM-ATTRIBUTES and /SHOW-VM-RESOURCES using the operand INFORMATION=*MONJV/*ALL as VM-MONJV.

Behavior when the system is abnormally terminated in the monitor system

MONJVs are retained in the monitor system if the system is terminated abnormally. They have the same status at first as was set **before** the system was terminated abnormally. When the monitor system is restarted, the MONJVs are updated as soon as possible.



If a VM in the computer network is monitored by a server or a VM outside the monitor VM, the program which is monitoring it must always take the network status of the monitor VM into consideration. HIPLEX MSCF also offers MONJVs to monitor the network status, see the “HIPLEX MSCF” manual [9], especially the section “Monitoring the network status”.

Without restarting the monitor system

MONJVs remain in the same status (with MONJV protection) as before the system was terminated abnormally. Changes to status cannot be displayed.

This MONJV can be used again in a later session with /CREATE-VM to monitor a VM.

With a restart of the monitor system

The MONJVs also remain valid in the session which is initiated by the restart.

If the status of the guest system or of the VM is changed during the restart phase, the MONJVs are updated as soon as the corresponding pubset can be accessed:

- the MONJVs are updated on the home pubset on `SYSTEM READY`.
- MONJVs on a data pubset are updated after the pubset has been imported. If the pubset is still not available after an hour, no further attempt is made to access the MONJVs. If the status of the guest system or the VM changes after this, VM2000 makes another attempt to update the relevant MONJV. For every attempt made to access a MONJV which fails, the message `VMS4343` is output at the main operator terminal.

Example

The following sample procedure can be used, for instance, to execute follow-up processing (e.g. data backup) after the guest systems have been shut down (e.g. using `/SHUTDOWN-VM VM-ID=*VM-GUESTS`).

```

/SET-PROC-OPT
/" *****
/" ** PROCEDURE VMCHECK *****
/" *****
/" ** THE PROCEDURE IS CALLED UP IN THE MONITOR SYSTEM. "
/" ** VMS NOT RUNNING ARE TERMINATED. "
/" ** WAITING FOR TERMINATION OF SYSTEM FOR RUNNING VMS. "
/" ** IMPORTANT STEPS: "
/" ** - CONNECTION TO VM2000 ESTABLISHED "
/" ** - CONFIGURED VMS CHECKED "
/" ** - VMS NOT RUNNING ARE TERMINATED "
/" ** - WAITING FOR TERMINATION OF SYSTEM IN GUEST SYSTEMS BY "
/" ** EVALUATING THE MONJV "
/" ** - DISCONNECTION FROM VM2000 "
/" *****
/" *****
/DECLARE-VARIABLE NAME=SHVMR(TYPE=STRUCTURE),MULT-E=*LIST
/DECLARE-VARIABLE NAME=TXT(TYPE=STRING,INIT-VALUE='OK')
/DECLARE-VARIABLE NAME=I(TYPE=INT)
/ASSIGN-SYSLST LST.VMCHECK
/MODIFY-JOB-OPTIONS LOG=*PAR(LIST=Y)
/MAIN: BEGIN-BLOCK DATA-INSERT=Y
/"** CONNECTION TO VM2000 "
/BEGIN-VM-DIALOG VM-ID=1

```



```

/ " ** CHECK VMS AND THEIR STATE "
/ EXEC-CMD (SHOW-VM-RESOURCES INF=*ALL),STRUCT-OUT=SHVMR
/ FOR I=*COUNT(FROM=2, TO= SIZE('SHVMR'), INCR=1 )
/ " ** TERMINATE VMS THAT ARE NOT RUNNING "
/ IF (SHVMR#I.STA NE '*RUN')
/   DELETE-VM VM-ID=&(SHVMR#I.VM-INDEX)
/   IF-CMD-ERROR
/     TXT = 'NO TERMINATION OF VM &(SHVMR#I.VM-INDEX)'
/     GOTO LABEL = ERROR
/   END-IF
/ ELSE      " ** WAIT FOR GUEST SYSTEM TERMINATION **"
/   WAIT-EVENT *JV(((&(SHVMR#I.VM-MONJV),1,2) <> '$R'),TIME-LIM=300)
/   IF-CMD-ERROR
/     TXT = 'NO SHUTDOWN IN VM &(SHVMR#I.VM-INDEX) '
/     GOTO LABEL = ERROR
/   END-IF
/   JVMON = JV(SHVMR#I.VM-MONJV)
/   IF NOT (SUBSTRING(JVMON,1,2) = '$D' AND SUBSTRING(JVMON,82,5) = 'NTERM')
/     TXT = 'VM &(SHVMR#I.VM-INDEX) NOT TERMINATED REGULARLY'
/     SEND-MSG TO=OPER,MSG='*** WARNING IN VMCHECK: &TXT ***'
/     TXT = 'NOT ALL VMS TERMINATED'
/   ELSE
/     DELETE-VM VM-ID=&(SHVMR#I.VM-INDEX)
/   END-IF
/ END-IF
/ END-FOR
/ END-BLOCK MAIN
/ IF-BLOCK-ERROR      "HANDLE OTHER ERRORS"
/   TXT = 'OTHER ERRORS IN THE SECTION'
/   GOTO LABEL = ERROR
/ END-IF
/ " ** NORMAL END "
/ ERROR:
/ END-VM-DIALOG VM-ID=1      "SHUT DOWN CONNECTION TO VM2000"
/ IF-CMD-ERROR; END-IF
/ IF (TXT = 'OK')
/   SEND-MSG TO=OPER,MSG='*** VMCHECK: ALL GUEST SYSTEMS TERMINATED ***'
/ ELSE
/   SEND-MSG TO=OPER,MSG='*** ERROR IN VMCHECK: &TXT ***'
/ END-IF
/ MOD-JOB-OPTIONS LOG=*PAR(LIST=NO)
/ ASSIGN-SYSLST *PRIMARY
/ END-PROC

```

5.2.3 Managing devices and assignment sets

After a VM has been created it can be assigned subsets and individual peripheral devices. To permit implicit assignment of subsets and devices by the guest system, the devices envisaged for this are grouped in what are known as assignment sets. The devices from the assignment sets for which a VM is privileged may be assigned (implicitly) to the VM by the guest system. See [section “Peripheral device management” on page 284](#).

5.2.4 VM and VM group

SQ VM groups are not available on SQ servers.

On S servers, VMs are added to a VM group by the VM2000 administrator when the VM is created using `/CREATE-VM (CPU-QUOTA=*BY-VM-GROUP operand)` or during ongoing operation using `/ADD-VM-TO-VM-GROUP`, see the [section “Adding a VM to a VM group” on page 179](#).

A VM is removed from a VM group implicitly by the VM administrator using `/DELETE-VM` or by the VM2000 administrator using `/REMOVE-VM-FROM-VM-GROUP`, see the [section “Removing a VM from a VM group” on page 180](#).

5.2.5 Assigning a VM or a VM group to a CPU pool

The VM2000 administrator assigns VMs which do not belong to a VM group to a CPU pool using `/CREATE-VM (CPU-POOL operand)` when the VM is initialized, see [page 143](#). After initialization, the VM2000 administrator can assign a VM which is not assigned to a VM group to another CPU pool during ongoing operation using `/ASSIGN-VM-TO-CPU-POOL`, see the [section “Assigning a VM or VM group to a \(different\) CPU pool” on page 232](#).

On S servers, the VM2000 administrator assigns VM groups to a CPU pool when the VM group is initialized using `/CREATE-VM-GROUP (CPU-POOL operand)`, see [page 178](#). After initialization, the VM2000 administrator can assign the VM group (with all member VMs) to another CPU pool during ongoing operation using `/ASSIGN-VM-GROUP-TO-CPU-POOL`, see the [section “Assigning a VM or VM group to a \(different\) CPU pool” on page 232](#).

5.2.6 Starting VM administration

After the VM has been initialized, the VM administrator can start the administration dialog with his or her VM (ADMIN dialog from a privileged user task (recommended) or using \$VMCONS).

The procedure for establishing a connection and the structure of the ADMIN dialog is described in detail in [section “VM2000 Operating strategy” on page 88](#).

The dialog from a privileged user task or using \$VMCONS is opened with the command `/BEGIN-VM-DIALOG VM-ID=...,PASSWORD=...,TYPE=*STD/*ADMIN(...)`.

Up to 15 ADMIN dialogs can be conducted concurrently for each VM from different, privileged user tasks. In addition, an ADMIN dialog can be conducted for each VM via \$VMCONS.

If a password for dialog access was defined when the VM was initialized (`/CREATE-VM`), it must be specified when opening a dialog with `/BEGIN-VM-DIALOG`.

5.2.7 Starting the guest system on a VM

The guest system on a VM can be started by the VM2000 or VM administrator of the VM. Starting the guest system (initial program load, IPL) is initiated with `/START-VM` after creating the VM and assigning the necessary devices. The startup for the guest system on the VM can then be carried out as in native operation (see the “Introductory Guide to Systems Support” [2]).



How the monitor system is started is described in the [chapter “Initializing and terminating VM2000” on page 65](#).

Before starting the guest system on a VM, at least one console device (a KVP console device pair or, on S servers, a virtual console), the disk devices of the home pubset and the IPL device must be explicitly assigned to it.

If the VM still has implicitly assigned devices when the guest system is started, these devices are removed from the VM.

Exception:

disk devices with diagnostic IPL (`/START-VM DIAGNOSTIC-IPL=*YES`, see [page 290](#))



On SQ servers the guest system can also be started via the SQ manager or using the SVP functions of the KVP console which is assigned to the VM (see the manual “SQ Series Business Servers, Operation and Administration” [18]).

In order to start the guest system on the VM, the device with the IPL disk device must be specified at the **first** start.



For S servers this can also be a tape device.

The IPL device must be **exclusively** and **explicitly** assigned. When the guest system is started again, the IPL device used last can be used again with the new default value `IPL-UNIT=*STD`.

The following optional specifications can be made:

- mnemonic device names of the KVP console device pair
If the mnemonic device name for this VM is not specified at the first `/START-VM`, VM2000 searches for a KVP console device pair assigned to the VM.
 - If, on S servers, no KVP console device pair is assigned to the VM, the virtual console of the VM is used.
- IPL information.
This defines the type of system initialization to be used for initial program loading of the guest system (fast startup, dialog startup or automatic startup).
- □ Up to 8 bytes of transfer information for the guest system. This corresponds to the specification in the `PARMS` field of the corresponding SVP menu screen.

- whether a diagnostic IPL is to be executed. At the same time the user can specify whether or not the save area is to be overwritten.
 - whether main memory is to be cleared during initial program loading. The clearing process can be very time-consuming.
- [S]** For a VM with `IO-RESET=*YES`, VM2000 carries out a system reset at the time of `/START-VM` as with a firmware IPL.

The values for `IPL-UNIT`, `MAIN-CONSOLE` and (on SQ servers) the system name (= VM name) are retained for the next startup of the guest system until a VM is shut down. They provide the default values for the next startup.

State of the VM

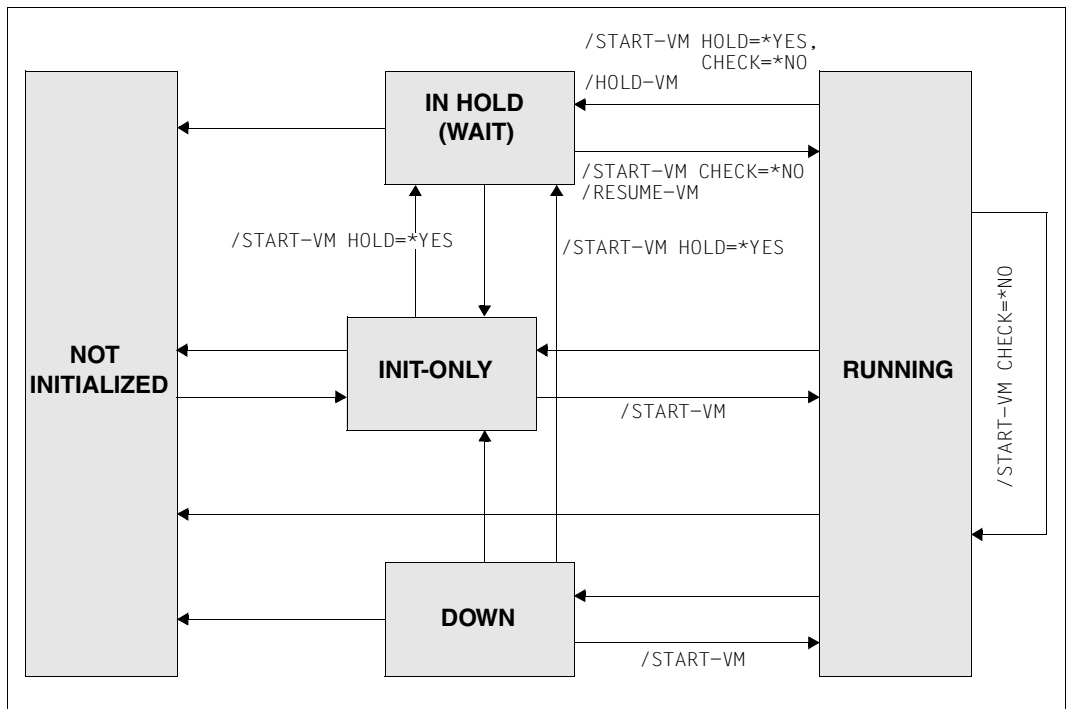
VM2000 prevents the accidental starting of a VM which is still active (`CHECK-VM-STATE=*YES`).

If a VM in the `IN HOLD (WAIT)` or `RUNNING` state is to be started, the operand `CHECK-VM-STATE=*NO` must be specified.

- [SQ]** A VM in the `MIGRATING` state should not be started. If it is to be restarted nevertheless in an exceptional case, the `CHECK-VM-STATE=*NO` operand must be specified.

The `HOLD` operand controls the state of the VM after successful execution of `/START-VM`. The following states are possible:

- `RUNNING` (operand `HOLD=*NO`)
 - [SQ]** Only the default value `HOLD=*NO` is allowed on SQ servers.
- `IN HOLD (WAIT)` (operand `HOLD=*YES`)

Figure 14: Changing the state of the VM by means of `/START-VM`

5.2.8 Information services

The information services supply information

- for the VM2000 administrator
 - on the distribution of VM2000 resources (/SHOW-VM-RESOURCES)
 - on the device status (/SHOW-VM-DEVICE-STATUS)
 - the assignment sets (/SHOW-VM-ASSIGNMENT-SET)
 - one-off or periodic monitored data for the overall VM2000 system (/SHOW-VM-STATUS)
 - on the attributes of CPU pools (/SHOW-VM-CPU-POOL)
 - on the attributes of VM groups (/SHOW-VM-GROUP)
- for the privileged VM administrator
 - on the distribution of VM2000 resources (/SHOW-VM-RESOURCES)
 - on the device status (/SHOW-VM-DEVICE-STATUS)
- for the VM administrator
 - on on the attributes of his/her VM (/SHOW-VM-ATTRIBUTES)

You will find examples of the VM2000 information commands and their outputs under the relevant command descriptions in the [chapter “VM2000 commands”](#).

VM2000 supplies output in S variables for all information commands, with the exception of /SHOW-VM-STATUS, when these commands are executed in a user task. The S variables can be evaluated with the software product SDF-P. For the structure of the S variables, see the relevant command descriptions in the [chapter “VM2000 commands”](#).

Information on the resources of VM2000

The attributes of the VMs, the status of the VM2000 resources and VM2000-specific information on the configuration of the server are shown with /SHOW-VM-RESOURCES, see [page 579](#).

This command is only permitted for the VM2000 administrator and VM administrators with ADMIN-PRIVILEGES=*MAXIMUM.

Information on the device status of VM2000

Using /SHOW-VM-DEVICE-STATUS you can display detailed information on the devices and their properties, see [page 556](#).

This command is only permitted for the VM2000 administrator and VM administrators with ADMIN-PRIVILEGES=*MAXIMUM.

Information on the assignment sets of VM2000

/SHOW-VM-ASSIGNMENT-SET outputs detailed information on the assignment sets and their assignment to the VMs, see [page 529](#).

This command is only permitted for the VM2000 administrator.

Information on the VM2000 load

/SHOW-VM-STATUS provides the VM2000 administrator with an overview of the current monitored data on active objects under VM2000, see [page 607](#).

The one-off output of monitored data (PERIOD=*RECENT) takes place synchronously to command execution and shows the monitored data from the immediate past. The one-off output can be called in parallel by multiple VM2000 administrators.

The periodic output of monitored data (PERIOD=<integer>) takes place asynchronously wherever the last /SHOW-VM-STATUS command was entered. Periodic output can only be used by one VM2000 administrator. An active /SHOW-VM-STATUS command with periodic output will be terminated if a new /SHOW-VM-STATUS PERIOD=*NO command is issued from the same input medium (ADMIN dialog of the monitor VM or BS2000 console), by the termination of the associated ADMIN dialog or by a shutdown.

Information on the attributes of CPU pools

/SHOW-VM-CPU-POOL outputs detailed information on the CPUs in a CPU pool and the VMs assigned to the CPU pool, see [page 549](#).

This command is only permitted for the VM2000 administrator.

Information on the attributes of VM groups (S servers)

/SHOW-VM-GROUP outputs detailed information on the attributes of the VM groups and the associated VMs, see [page 573](#).

This command is only permitted for the VM2000 administrator.

Information on the attributes of a VM

The VM attributes are output using /SHOW-VM-ATTRIBUTES, see [page 532](#).

This command is intended for use by the VM administrator.

5.2.9 Resource management

Resource management during VM2000 operation comprises management of the

- modifiable VM attributes

Using `/MODIFY-VM-ATTRIBUTES`, the VM2000 administrator and VM administrators with `ADMIN-PRIVILEGES=*MAXIMUM` (see [page 193](#)) can modify the attributes of a VM which were defined during VM initialization (`/CREATE-VM`).

The following changes can be made:

- the CPU quota and the maximum CPU utilization
 - the maximum IO utilization
 - the password for administration (`/BEGIN-VM-DIALOG`)
 - the range of commands available to the VM2000 and VM administrators. The range can be limited (VM2000 administrator) or extended (VM administrator), see [page 192](#)
 - the privileges of the VM, see [page 145](#)
 - the scope of the trace entries to be logged
- peripheral devices for a VM and for the VM2000 devices (see [page 284](#)).

The devices and pubsets can be explicitly managed using the following commands:

- `/ADD-VM-DEVICES` (to attach devices to a VM)
- `/REMOVE-VM-DEVICES` (to detach devices from a VM)
- `/SWITCH-VM-DEVICES` (to detach devices from one VM and attach them to another VM).

To permit implicit assignment of devices and pubsets by the guest system, the devices envisaged for this are grouped in what are known as assignment sets. The devices from the assignment sets for which a VM is privileged may be assigned (implicitly) to the VM by the guest system.

Special device properties and the assignment and usage modes can be modified using `/MODIFY-VM-DEVICE-ATTRIBUTES` and `/MODIFY-VM-DEVICE-USAGE`.

- reconfiguration of the main memory for the VM (see [page 270](#))
The main memory for a VM can be enlarged with `/EXTEND-VM-MEMORY` and reduced with `/REDUCE-VM-MEMORY`.
S On S servers, free main memory areas can also be extended or reduced by moving a VM with `/MOVE-VM` (see [page 166](#)).
- assignment of global storage to a VM (S servers, see [page 277](#))
Global storage can be assigned to the VMs with `/ADD-VM-RESOURCES` and the assignment canceled with `/REMOVE-VM-RESOURCES`.

- real CPUs which are available for VM2000 operation (see [page 218](#))
These can be attached using `/ATTACH-VM-RESOURCES` and detached using `/DETACH-VM-RESOURCES`.
- (dynamic) CPU pools (see [page 225](#))
The commands defined here can be used to combine real CPUs into disjunctive CPU pools. Virtual machines can be assigned to the CPU pools.

5.2.10 Suspending a VM/resuming operation

A VM in the RUNNING state can be suspended using `/HOLD-VM`. `/RESUME-VM` cancels a previously initiated wait state.



Precautions must be taken for guest systems in the MSCF network to ensure that the guest system is not regarded as failed while it is suspended.



On SQ servers `/HOLD-VM` does not suspend the firmware components of the VM. The VM can still consume a minimal amount of CPU performance.

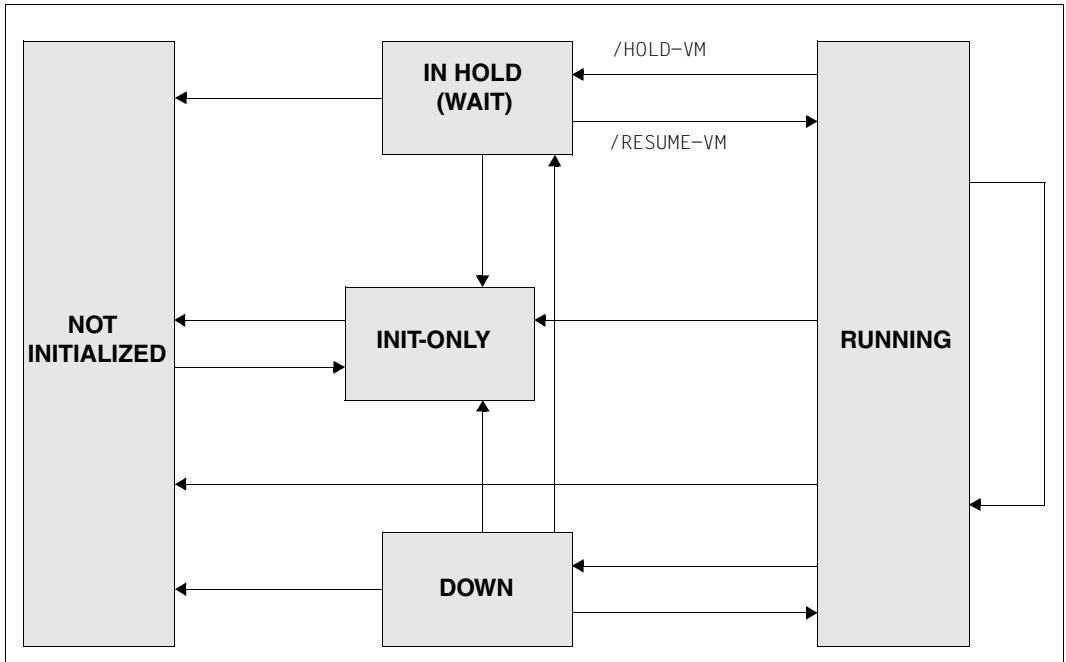


Figure 15: Changing the VM state by means of `/HOLD-VM` and `/RESUME-VM`

There are three different wait states:

- **WAIT (VMA)** initiated by a `/HOLD-VM` from the VM administrator (operand `VM-ID=*CURRENT`). The VM with which the dialog was started using `/BEGIN-VM-DIALOG` is suspended.
- **WAIT (SEL)** initiated by a **selective** `/HOLD-VM` from the VM2000 administrator (operand `VM-ID=2..n`). The VM with the specified VM identification is suspended.
- **WAIT (GLB)** initiated by a **global** `/HOLD-VM` from the VM2000 administrator. (operand `VM-ID=*ALL`). All currently executing VMs (except the monitor VM) are suspended.

These states may also overlap. For instance, a VM can be placed in the `/HOLD-VM` state by the VM administrator and in the global `/HOLD-VM` state by the VM2000 administrator.

The VM administrator can display the wait states of his/her VM using `/SHOW-VM-ATTRIBUTES`. `/SHOW-VM-RESOURCES` displays the states of all VMs for the VM2000 administrator.

The following table shows the possible changes of state to which a VM can be subjected as a result of suspension.

Command /HOLD-VM	VM2000 administrator global (VM-ID=*ALL)	VM2000 administrator selective (VM-ID=2..n)	VM administrator (VM-ID=*CURRENT)
Old status			
RUNNING	GLB	SEL	VMA
VMA	VMA+GLB	VMA+SEL	VMA
SEL	SEL+GLB	SEL	SEL+VMA
GLB	GLB	GLB+SEL	GLB+VMA
VMA+SEL	VMA+SEL+GLB	VMA+SEL	VMA+SEL
VMA+GLB	VMA+GLB	VMA+GLB+SEL	VMA+GLB
SEL+GLB	SEL+GLB	SEL+GLB	SEL+GLB+VMA
VMA+SEL+GLB	VMA+SEL+GLB	VMA+SEL+GLB	VMA+SEL+GLB

Key

RUNNING VM is running.

GLB Global suspension by the VM2000 administrator (VM-ID=*ALL).

SEL Selective suspension by the VM2000 administrator (VM-ID=2..n).

VMA Suspension by the VM administrator (VM-ID=*CURRENT).

Changing states using the RESUME-VM command

The following rules must be observed when canceling a wait state using /RESUME-VM:

- A **selective** /RESUME-VM command from the VM2000 administrator (operand VM-ID=2..n) always imposes the RUNNING state. This cancels all VM wait states.
- A **global** /RESUME-VM command from the VM2000 administrator (operand VM-ID=*ALL) cancels a global /HOLD-VM command.
- /RESUME-VM from the VM administrator (operand VM-ID=*CURRENT) cancels a /HOLD-VM from the VM administrator.

The following table shows the possible changes of state to which a VM can be subjected by means of /RESUME-VM:

Command /RESUME-VM	VM2000 administrator global (VM-ID=*ALL)	VM2000 administrator selective (VM-ID=2..n)	VM administrator (VM-ID=*CURRENT)
Old status			
RUNNING	RUNNING	RUNNING	RUNNING
VMA	VMA	RUNNING	RUNNING
SEL	SEL	RUNNING	SEL
GLB	RUNNING	RUNNING	GLB
VMA+SEL	VMA+SEL	RUNNING	SEL
VMA+GLB	VMA	RUNNING	GLB
SEL+GLB	SEL	RUNNING	SEL+GLB
VMA+SEL+GLB	VMA+SEL	RUNNING	SEL+GLB

Example

VM2000 administrator	VM administrator of VM2	VM2 status
	/BEGIN-VM-DIALOG VM-ID=2	
	/START-VM VM-ID=2	RUNNING
	/HOLD-VM VM-ID=*CURRENT	VMA
/HOLD-VM VM-ID=*ALL		VMA+GLB
	/RESUME-VM VM-ID=*CURRENT	GLB
/RESUME-VM VM-ID=2		RUNNING
/HOLD-VM VM-ID=*ALL		GLB
/RESUME-VM VM-ID=*ALL		RUNNING

5.2.11 Moving a VM within main memory

 On SQ servers the location of a VM need not be changed. `/MOVE-VM` is not available.

`/MOVE-VM` enables the VM2000 administrator on S servers to (retrospectively) change the location of a VM in main memory from the original location defined during initialization.

The new location may overlap the old VM main memory. The new address must be a multiple of 1 Mbyte.

It is possible to protect a guest system BS2000/OSD as of V7.0 in the MSCF network from being regarded as failed by MSCF partners while it is being moved.



For guest systems BS2000/OSD < V7.0 for which no automatic protection takes place via MSCF, precautions must be taken (as in earlier VM2000 versions) to ensure that the guest system is not regarded as failed while it is being moved.

5.2.12 Passing information to VM administrators

Messages can be sent to the VM administrators by means of `/SEND-VM-MESSAGE`, e.g. informing him/her of monitor system shutdown. The message can be directed either to all VM administrators or to a selected VM administrator.

The message text is delivered to the addressed administrators as an insert to the asynchronous message `VMS2036`.

5.2.13 Modifying the VMCONS parameters

`/MODIFY-VMCONS-PARAMETERS` enables the following \$VMCONS parameters which were set in the parameter file of the VM2-MON subsystem to be modified:

- the OPERID password for the UCON logon of \$VMCONS
- the password for establishing the connection to \$VMCONS

The new passwords are valid for new connections after the command has been executed. Existing connections continue to run unchanged.

The new passwords remain valid until the end of the current session or until the command is entered again. They also remain valid after a restart of \$VMCONS. However, they are no longer available after an automatic restart of the monitor system. Changes for the next session must be made in the parameter file of the VM2-MON subsystem.

5.2.14 Terminating VM administration

You terminate an ADMIN dialog from a privileged user task or via \$VMCONS with the command `/END-VM-DIALOG VM-ID=...,TYPE=*ADMIN`.

When the last (or only) dialog of a connection to \$VMCONS is terminated, you can use the `DISCONNECT` operand to specify whether the connection to \$VMCONS is to be cleared or maintained.

`/END-VM-DIALOG` also clears the connection to \$VMCONS if this is the only connection, i.e. if no dialog has yet been started.

The VM administrator can enter `/END-VM-DIALOG` for his/her VM.

If the VM2000 administrator enters `/END-VM-DIALOG` for a VM other than the monitor VM, all affected dialogs of the VM are terminated, depending on the specification in the `TYPE` operand.

If the VM2000 administrator enters `/END-VM-DIALOG` for the monitor VM **from a BS2000 console**, all ADMIN dialogs of the monitor VM are terminated.

If a VM2000 administrator terminates the ADMIN dialog with the monitor VM, any active `/SHOW-VM-STATUS` command with periodic output from this VM2000 administrator is terminated.

5.2.15 Aborting the guest system on a VM

`/RESET-VM` aborts the guest system on a VM, i.e. the virtual CPUs of the VM are stopped.

The attributes and resources of the VM remain intact. ADMIN and VC dialogs continue to run. Implicitly assigned devices (see [page 288](#)) are, however, removed from the VM. If main memory reconfiguration is being carried out, it is terminated.

[S] On S servers the Channel Subsystem Monitoring Facility function is switched off for the guest system.

The following tasks are possible with `/RESET-VM`:

- suppression of disk accesses by a guest system that has apparently failed
- reduction of `MEMORY-SIZE` with `/REDUCE-VM-MEMORY`
(if problems occur at command execution during operation)
- reduction of `MIN-MEMORY-SIZE` with `/REDUCE-VM-MEMORY` (S servers)

[SQ] On SQ servers the “Abort VM” function of the SQ manager also aborts the guest system, see the manual “SQ Series Business Servers, Operation and Administration” [18].

State of the VM

After execution of `/RESET-VM`, the VM adopts the INIT-ONLY state.

[SQ] A VM in the MIGRATING state cannot be aborted.

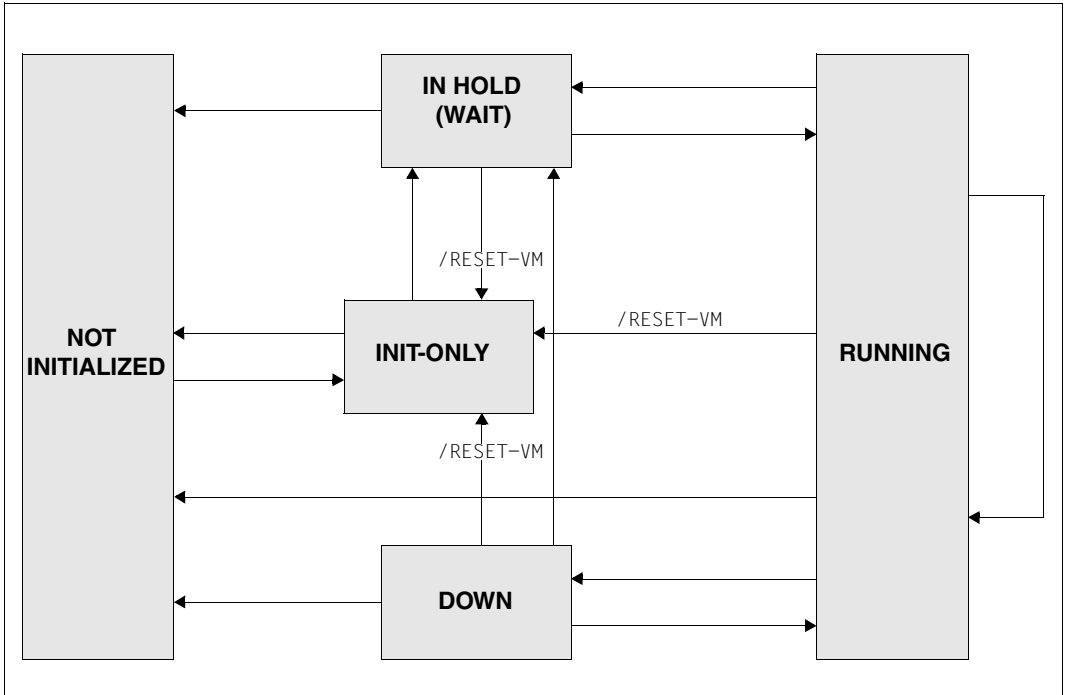


Figure 16: Changing the state of the VM by means of /RESET-VM

5.2.16 Shutdown in the guest system and of VM2000

`/SHUTDOWN-VM` executes a shutdown (without a restart) in the selected BS2000 guest systems as of BS2000/OSD V7.0:

- The VM administrator uses the `VM-ID=<vm-id>` operand to shut down the guest system on his/her VM. The VM2000 administrator can use this to shut down any guest system with the exception of the monitor system.
- The VM2000 administrator uses the `VM-ID=*VM-GUESTS` operand to shut down the guest systems of all VMs with the exception of the monitor system.
- The VM2000 administrator uses the `VM-ID=*VM2000` operand to shut down all VMs and last of all the monitor system (coordinated shutdown of VM2000).

Shutdown for a guest system

The shutdown for a BS2000 guest system (as of BS2000/OSD V7.0), with the exception of the monitor system, is initiated by the VM or VM2000 administrator using `/SHUTDOWN-VM . . . ,ACTION=*ENTER-JOB / *SHUTDOWN(. . .) .`

`ACTION=*ENTER-JOB` initiates the shutdown in the guest system by calling the `$TSSO.SYS.VM.SHUTDOWN` command file as an enter job. The command file permits customer specific preparations to be made for the shutdown, e.g. termination of particular applications. The last action of the customer specific preparations must be the BS2000 command `/SHUTDOWN MODE=*NQ / *END(RESTART=*NQ)`. The privilege OPERATING is required for the execution of `/SHUTDOWN` (see [page 92](#)).

`ACTION=*SHUTDOWN(QUIET-TIME=. . .)` initiates the shutdown in the guest system by calling shutdown processing (analogously to the BS2000 command `/SHUTDOWN`). Here the time which the guest system remains in QUIET state (`/SHUTDOWN MODE=*QUIET`) before it is shut down (`/SHUTDOWN MODE=*NQ / *END(RESTART=*NQ)`) is specified in minutes.

- SQ On SQ servers the shutdown can also be initiated for a guest system via the SQ manager, see the manual “SQ Series Business Servers, Operation and Administration” [18].

Coordinated shutdown of VM2000

Coordinated shutdown of VM2000 is initiated by the VM2000 administrator and executed in two stages:

1. For all BS2000 guest systems (as of BS2000/OSD V7.0) with the exception of the monitor system, the shutdown is initiated as described above.

i No action is initiated for other guest systems. These must be shut down in advance by the operator concerned. Otherwise the VM is terminated when shutdown is executed for the monitor system regardless of the current guest system state (see [page 83](#)).

The monitor system waits for system shutdown in the guest systems. The maximum wait time here is specified in the `WAITING-TIME` operand. The wait time begins with the start of the enter jobs or when shutdown processing is initiated (`MODE=*END`) in the guest systems.

i The maximum wait time selected should be greater than the longest `SHUTCTL` wait time in the BS2000 guest systems (as of BS2000/OSD V7.0), see system parameter `SHUTCTL` in the “Introductory Guide to Systems Support” [2]. This enables the guest systems to be terminated before the monitor system is shut down.

While shutdown is being executed in the guest systems, the monitor system continues to operate normally. VM2000 commands can still be entered. However, the following commands are rejected with the message `VMS2101`:

```
- /BEGIN-VM-DIALOG TYPE=*ADMIN/*BOTH
- /CREATE-VM
- /HOLD-VM (for a VM administrator)
- /SHUTDOWN-VM
- /START-VM . . . ,DIAGNOSTIC-IPL=*NO (“normal” IPL)
```

S On S servers VM2000 also prevents a guest system from restarting.

2. When all guest systems have been shut down (all virtual machines are in the `INIT-ONLY` or `DOWN` state) or after the wait time has elapsed, the monitor system performs the shutdown itself as described above. All other guest systems are shut down implicitly here by means of `/DELETE-VM CHECK-VM-STATE=*NO` regardless of their state.

S On S servers operation of VM2000 and of the BS2000/OSD server is terminated when the monitor system shuts down.

SQ On SQ servers the server unit of the SQ server still remains in operation, but it can be shut down via the SQ manger.

An example of coordinated shutdown is provided on [page 81](#).

A new IPL can then take place on all server series.

Notes on shutdown processing using /SHUTDOWN-VM

- Active backup runs are ignored as if the class-2 system parameter SHUTARCH=N were set in the monitor system and in all the guest systems, see the “Introductory Guide to Systems Support” [2].
- If shutdown processing has already been initiated for a VM using /SHUTDOWN-VM VM-ID=<vm-id>/*VM-GUESTS, ACTION=*ENTER-JOB or *SHUTDOWN(. . .) with a significant QUIET-TIME, shutdown processing can then be initiated immediately (MODE=*END) by entering /SHUTDOWN-VM VM-ID=<vm-id>/*VM-GUESTS, ACTION=*SHUTDOWN(QUIET-TIME=0).
- After shutdown processing has been initiated using /SHUTDOWN-VM VM-ID=*VM2000(. . .), VM2000 prevents the monitor system or a guest system from restarting. However, SLED can still be executed.
 - SQ On SQ servers VM2000 **cannot** prevent the guest system from being started or restarted via the SQ manager or by means of the SVP functions of the KVP console which is assigned to the VM.
- When /SHUTDOWN-VM VM-ID=<vm-id>/*VM-GUESTS is followed by a coordinated shutdown of VM2000 by means of /SHUTDOWN-VM VM-ID=*VM2000(. . .) and a shorter QUIET-TIME or QUIET-TIME=0, the QUIET-TIME entered beforehand for the guest system will be reduced accordingly.
- /SHUTDOWN-VM VM-ID=*VM2000(. . .), ACTION=*ENTER-JOB is rejected if /SHUTDOWN-VM VM-ID=<vm-id>/*VM-GUESTS, ACTION=*SHUTDOWN was entered beforehand.
 - SQ /SHUTDOWN-VM VM-ID=<vm-id>/*VM2000 is rejected for a VM in the MIGRATING state.
 - /SHUTDOWN-VM VM-ID=*VM-GUESTS is not executed for a VM in the MIGRATING state.

State of the VM

After `/SHUTDOWN-VM` has been executed in the guest system, the VM concerned assumes the **DOWN** state. This state is also assumed after `/SHUTDOWN` or abnormal system termination (**SETS**) in the guest system.

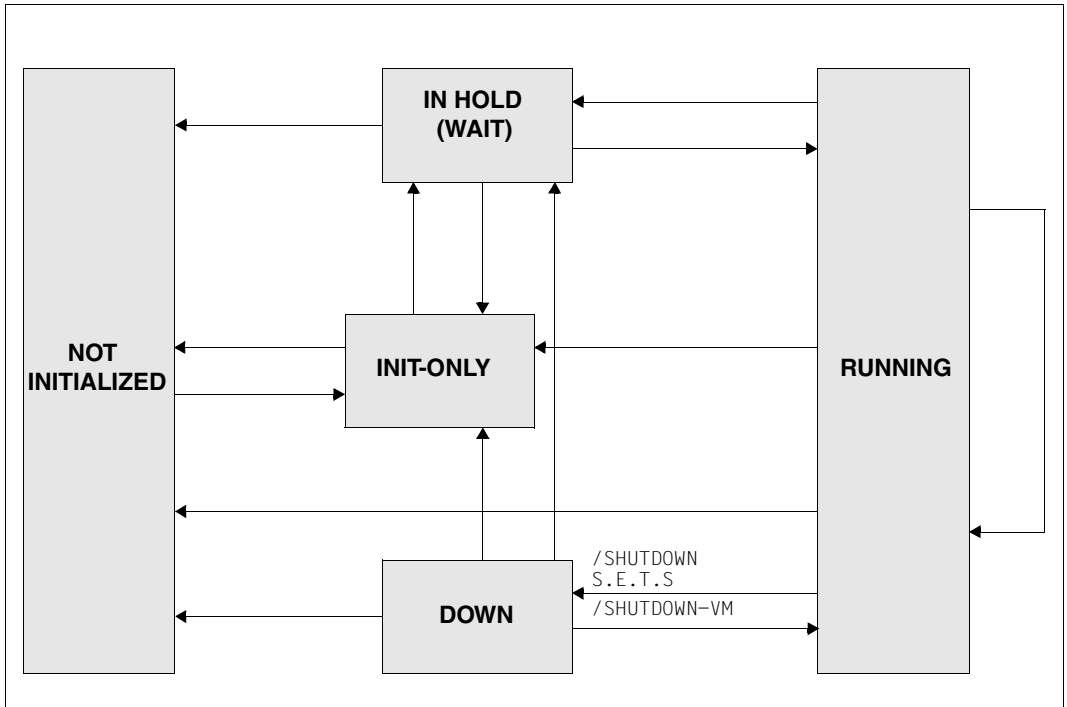


Figure 17: Changing the state of a VM by means of shutdown

5.2.17 Terminating a VM

A VM is terminated with `/DELETE-VM`. The main memory assignment, the device assignment for all devices, the assignment to a CPU pool, the affiliation to a VM group and the initialization of the VM are then cleared down.

[SQ] On SQ servers the main memory is released asynchronously and this can take several seconds. Only then is the released memory available for other or new VMs.

`/DELETE-VM` can be issued by the VM administrator for his/her VM. The VM2000 administrator can also terminate all existing VMs, except the monitor VM, with this command.

If `/DELETE-VM` is specified, VM2000 terminates all ADMIN dialogs with the VM and all VC dialogs with the guest system on the VM.

The output of periodic monitored data (`/SHOW-VM-STATUS PERIOD=<integer>`) is terminated for this VM.

Effects on the guest system

`/DELETE-VM . . . ,CHECK-VM-STATE=*NO` terminates an active guest system immediately. As a result, files of the guest system which is running on the VM may be inconsistent. To prevent this from happening, a shutdown should be executed before the VM in the guest system is terminated.

State of the VM

VM2000 prevents the accidental termination of an active VM (`CHECK-VM-STATE=*YES`). If a VM in the IN HOLD (WAIT) or RUNNING state is to be terminated, the operand `CHECK-VM-STATE=*NO` must be specified.

[SQ] A VM in the MIGRATING state should not be terminated. If in an exceptional case it should nevertheless be terminated, the `CHECK-VM-STATE=*NO` operand must be specified.

After `/DELETE-VM` has been executed, the VM is in the NOT INITIALIZED state.

Executing `/DELETE-VM` implicitly executes `/HOLD-VM` first.

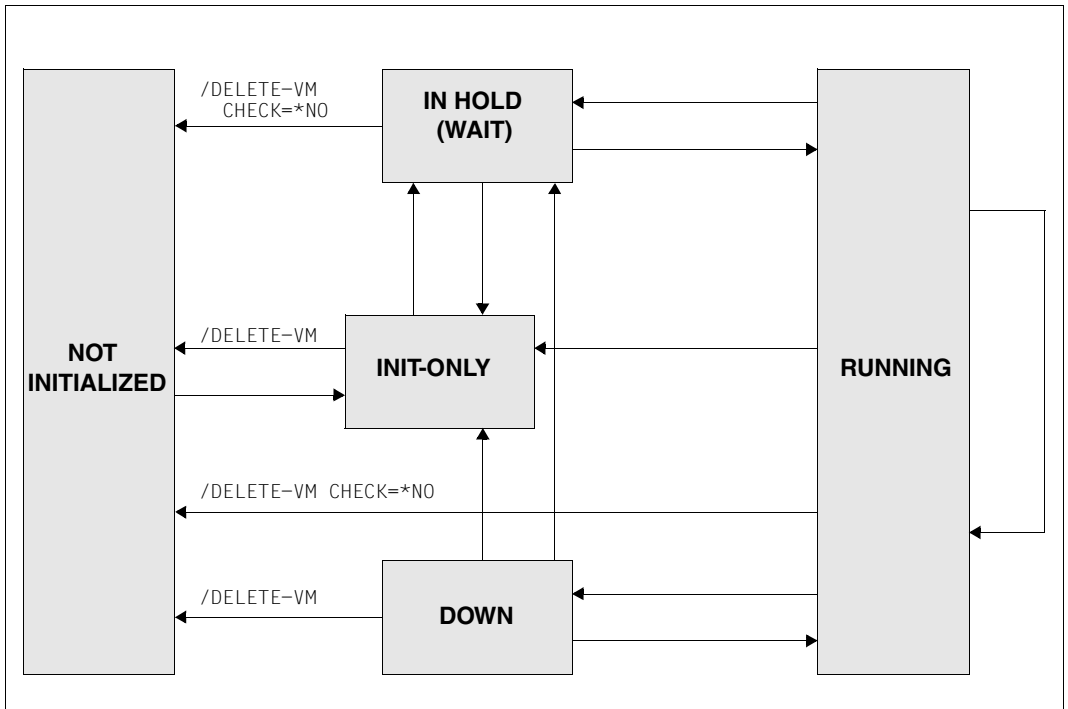


Figure 18: Changing the state of the VM by means of /DELETE-VM

5.3 Operation of VM groups (S servers)

The VM2000 concept of VM groups enables the VM2000 administrator to combine multiple VMs in an organizational unit. VM groups have attributes of their own and are assigned to a CPU pool. The VMs which belong to the VM group are taken into account in a special way for scheduling.

SQ VM groups are not available on SQ servers.

Attributes of VM groups and their members

VM groups have their own CPU quota (`CPU-QUOTA` operand). This determines the VM group's share of the CPU capacity of the CPU pool in comparison to other VM groups located in the CPU pool and VMs which do not belong to a VM group. Similarly to the virtual machine, the CPU utilization of the VM group can be restricted by the maximum CPU utilization (`MAX-CPU-UTILIZATION` operand).

For scheduling, too, the VM group is regarded as a unit. If a VM in the VM group has not yet used up the CPU share intended for it, the other VMs of this VM group are given preference over the other VMs of the CPU pool during scheduling. This is also referred to as “load balancing within a VM group”.

The VMs which belong to a VM group are also referred to as “members of the VM group”. They have their own member CPU quota (`MEMBER-CPU-QUOTA` operand) which determines the CPU share of the VM compared to the other VMs in the VM group. The maximum CPU utilization of a member VM is restricted by both the “maximum CPU utilization of the VM” and the “maximum CPU utilization of the VM group”.

Further information on the CPU quota, maximum CPU utilization and scheduling is provided in the sections [“Planning distribution of the CPU capacity to the VMs” on page 238](#) and [“Scheduling procedures” on page 249](#).

The monitor VM can also be added to a VM group.

Commands for operating VM groups

The VM2000 administrator manages a VM group in the ADMIN dialog using the monitor VM. The function and privileges of the VM2000 administrator are described in the [section “Operating the whole VM2000 system” on page 186](#).

The table below shows the individual operation groups with the associated VM2000 commands for operating a VM group.

Operation group	VM2000 command
Creating a VM group	CREATE-VM-GROUP ¹
Adding a VM to a VM group	CREATE-VM ¹ ADD-VM-TO-VM-GROUP ¹
Modifying the attributes of a VM in a VM group	MODIFY-VM-ATTRIBUTES ²
Removing a VM from a VM group	DELETE-VM ³ REMOVE-VM-FROM-VM-GROUP ¹
Modifying the attributes of a VM group	MODIFY-VM-GROUP-ATTRIBUTES ¹ ASSIGN-VM-GROUP-TO-CPU-POOL ¹
Outputting information on a VM group	SHOW-VM-GROUP ¹
Deleting a VM group	DELETE-VM-GROUP ¹

¹ For use by the VM2000 administrator only

² Command for the VM2000 administrator and those VM administrators with ADMIN-PRIVILEGES=*MAXIMUM (see [page 193](#))

³ For use by the VM2000 administrator and the VM administrator of the VM

These operation groups are described in more detail in the sections below.

5.3.1 Creating a VM group

Before a VM group can be worked with, it must be created in VM2000. A VM group is created by the VM2000 administrator using `/CREATE-VM-GROUP`.

When a VM group is created it is assigned the following attributes:

- Name of the VM group
- CPU quota and maximum CPU utilization of the VM group
- CPU pool of the VM group

Up to half as many VM groups as VMs may be created (rounded up). The number of VMs is hardware-dependent, see [page 457](#).

Example

A maximum of 15 VMs are permitted on an S server. A maximum of 8 VM groups can thus be created.

The maximum number of VMs and the maximum number of VM groups are also displayed using `/SHOW-VM-RESOURCES INFORMATION=*CONFIGURATION`.

Name of the VM group

The name of a VM group can be freely selected. It must identify the VM group unambiguously. `/CREATE-VM-GROUP` is rejected if a specified VM group name has already been assigned.

Recommendations for defining and using the names of VM groups

The name of a VM group should be unique within a VM2000 installation (data center organization) and indicate the purpose of the VM group.

The name ranges for VMs, VM groups and CPU pools should be disjunctive.

CPU quota and maximum CPU utilization of the VM group

The CPU quota of the VM group determines the share of the VM group in the CPU capacity of the assigned CPU pool in comparison to the other VMs which are assigned to the same CPU pool either directly or as members of a VM group.

The CPU utilization of a VM group can be restricted by the maximum CPU utilization. This value also implicitly restricts the CPU utilization of every VM which belongs to the VM group.

The CPU quota and maximum CPU utilization of the VM group can be modified later using `/MODIFY-VM-GROUP-ATTRIBUTES`.

CPU pool, to which the VM group is assigned

Each VM group is always assigned to precisely one CPU pool. By default (CPU-Pool-NAME=*STDPool operand) the VM group is assigned to the standard CPU pool at initialization time.

The assignment of a VM group to a CPU pool can be changed later using /ASSIGN-VM-GROUP-TO-CPU-POOL.

5.3.2 Adding a VM to a VM group

After a VM group has been created, VMs can be added to this VM group. The VM2000 administrator adds VMs to a VM group when the VM is created using /CREATE-VM (CPU-QUOTA=*BY-VM-GROUP operand) or in ongoing operation using /ADD-VM-TO-VM-GROUP.

When this is done in ongoing operation, the VM may not already belong to a VM group. It must be assigned to the CPU pool to which the VM group is assigned.

When a VM is added to a VM group, a **member CPU quota** is specified for it which determines the VM's own CPU share in comparison to the other VMs in the VM group. The CPU quota of the (autonomous) VM is relinquished.

The maximum CPU utilization of a member VM is restricted by both the "maximum CPU utilization of the VM" and the "maximum CPU utilization of the VM group". The **maximum CPU utilization** of a VM is determined when the VM is created using /CREATE-VM. It can be modified later using /MODIFY-VM-ATTRIBUTES.

5.3.3 Modifying the attributes of a VM in a VM group

In addition to the general attributes of a VM, the following attributes which are important for scheduling within a VM group can also be modified.

The VM2000 administrator can modify the **member CPU quota** of a VM in a VM group using `/MODIFY-VM-ATTRIBUTES (CPU-QUOTA=*BY-VM-GROUP(MEMBER-CPU-QUOTA=...))` operand). As a result the VM's own CPU share is modified in comparison to the other VMs in the VM group.

The VM2000 administrator can also modify the **maximum CPU utilization** of a VM in a VM group using `/MODIFY-VM-ATTRIBUTES (MAX-CPU-UTILIZATION operand)`. This value is implicitly restricted by the maximum CPU utilization of the VM group.

5.3.4 Removing a VM from a VM group

A VM is removed from a VM group by the VM2000 administrator using `/REMOVE-VM-FROM-VM-GROUP` or implicitly by the VM administrator using `/DELETE-VM`.

When a VM is removed from a VM group using `/REMOVE-VM-FROM-VM-GROUP`, the VM which then becomes autonomous is assigned a new CPU quota. The VM's member CPU quota is relinquished.

When the last VM is removed from a VM group, the share of the CPU capacity of the CPU pool changes for the remaining VM groups and VMs which are assigned to the same CPU pool either directly or as members of a VM group.

5.3.5 Modifying the attributes of a VM group

The VM2000 administrator can modify the **CPU quota** and **maximum CPU utilization** of a VM group using `/MODIFY-VM-GROUP-ATTRIBUTES`.

As a result the CPU share of the VM group is modified in comparison to the other VM groups and VMs which are assigned to the same CPU pool either directly or as members of a VM group.

The VM2000 administrator can assign the VM group to another **CPU pool** using `/ASSIGN-VM-GROUP-TO-CPU-POOL`.

5.3.6 Outputting information on a VM group

The VM2000 administrator obtains information on a VM group using `/SHOW-VM-GROUP`.

The VM2000 administrator obtains the following information with this command:

- The attributes of the VM group as defined with `/CREATE-VM-GROUP` or `/MODIFY-VM-GROUP-ATTRIBUTES`
- The attributes of the VMs which belong to the VM group

The VM2000 administrator also obtains information on which VM group a VM belongs to using `/SHOW-VM-RESOURCES`, and the VM administrator using `/SHOW-VM-ATTRIBUTES`.

5.3.7 Deleting a VM group

After all VMs have been removed from a VM group, the VM group can be deleted. The VM2000 administrator deletes a VM group using `/DELETE-VM-GROUP`.

5.3.8 Example of operating VM groups

The example below illustrates how the VM2000 administrator operates a VM group with two VMs on an S server.

```

/show-vm-resources information=*std _____ (1)
          CPU-POOL(P)
VM-ID      VM-GROUP(G)  CPU-Q   MAX-U   PROC   MEM-SIZE  MAX-IO
  1 MONITOR  *STDPOOL(P)  30.00  100.00  2(+1)   128 MB (100)
  2 C1SYS1   *STDPOOL(P)  20.00   20.00  2(+1)   128 MB (100)

VM-ID      STATE          GUEST   VERSION ADMIN PRIV
  1 MONITOR  RUNNING          BS2-OSD V09.0A YES  AS
  2 C1SYS1   RUNNING          BS2-OSD V09.0A NO   AS
F R E E   VM:13          LARGEST-DOMAIN: 3814 MB
/create-vm-group vm-group-name=gcust1,cpu-quota=20,
                max-cpu-utilization=20,cpu-pool-name=*stdpool _____ (2)
VMS4401 VM GROUP 'GCUST1' CREATED
VMS4625 VM GROUP 'GCUST1' ASSIGNED TO CPU POOL '*STDPOOL'
/add-vm-to-vm-group vm-id=c1sys1,vm-group-name=gcust1,
                   member-cpu-quota=1 _____ (3)
VMS4425 VIRTUAL MACHINE (2,C1SYS1) ASSIGNED TO VM GROUP 'GCUST1'

```

- (1) The command shows the initial situation for this example:
A customer has agreed on a certain CPU capacity with the data center which corresponds to 20% of the CPU capacity on this server. The VM2000 administrator has therefore equipped the customer VM `C1SYS1` with a CPU quota and a maximum CPU utilization of 20. The monitor and customer VMs run under BS2000/OSD V9.0. The VM2000 administrator also operates the customer VM. The meaning of the output columns is described in more detail in the [section “Planning distribution of the CPU capacity to the VMs” on page 238](#).
- (2) The customer would now like to use a second VM. However, the specified CPU capacity should remain unchanged. For this purpose the VM2000 administrator creates the VM group `GCUST1` and assigns it a CPU quota and a maximum CPU utilization of 20. The VM group is assigned to the standard CPU pool.
- (3) The customer VM `C1SYS1` is added to the VM group `GCUST1`. The member CPU quota selected is 1; the CPU quota which the VM was given when it was created (20) is relinquished.

```

/creates-vm memory-size=128,vm-name=c1sys2,processor=*bi,
  cpu-quota=*by-vm-group(vm-group-name=gcust1,member-cpu-quota=1) ----- (4)
VMS3050 VIRTUAL MACHINE (3,C1SYS2) CREATED
VMS4425 VIRTUAL MACHINE (3,C1SYS2) ASSIGNED TO VM GROUP 'GCUST1'
VMS4615 VIRTUAL MACHINE (3,C1SYS2) ASSIGNED TO CPU POOL '*STDPOOL'
/show-vm-group vm-group-name=gcust1,information=*all ----- (5)
VM-GROUP CPU-POOL CPU-Q MAX-U
GCUST1 *STDPOOL 20.00 20.00

VM-GROUP CPU-POOL VM-ID CPU-Q / EFF-Q MAX-U / EFF-U
GCUST1 *STDPOOL 2 C1SYS1 1.00 / 10.00 20.00 / 20.00
          3 C1SYS2 1.00 / 10.00 100.00 / 20.00

...

/modify-vm-group-attributes vm-group-name=gcust1,cpu-quota=30,
  max-cpu-utilization=30 ----- (6)
/show-vm-group vm-group-name=gcust1,information=*attributes
VM-GROUP CPU-POOL CPU-Q MAX-U
GCUST1 *STDPOOL 30.00 30.00
/modify-vm-attributes vm-id=c1sys1,
  cpu-quota=*by-vm-group(member-cpu-quota=2),max-cpu-utilization=30 --- (7)

```

- (4) The customer VM `C1SYS2` is created. It is added to the VM group `GCUST1` when it is created (and consequently to the standard CPU pool) and is also assigned a member CPU quota of 1.
- (5) The information output for the VM group shows the status now achieved: The two customer VMs `C1SYS1` and `C1SYS2` share the specified CPU capacity (same CPU quota, i.e. 10% each (planned values)). The maximum CPU utilization of `C1SYS2` was set to 100 by the default value in the command, but is reduced to 20 by the relevant value for the group.
- (6) After a certain time the customer requires higher CPU capacity and agrees with the data center on a CPU capacity of 30% of the server capacity. For this purpose the VM2000 administrator increases the relevant values for the VM group `GCUST1`.
- (7) In addition, the defined CPU capacity should be distributed to the customer VMs in such a way that `C1SYS1` is assigned twice as much CPU capacity as `C1SYS2`. For this purpose the VM2000 administrator doubles the CPU quota of the VM `CSYS1` and increases its maximum CPU utilization (from 20) to the specified share (30).

```

/show-vm-resources information=*cpu,
                               vm-id=*by-vm-group(vm-group-name=gcust1) _____ (8)
VM-ID      CPU-POOL  GR-CPU-Q  CPU-Q/  EFF-Q  MAX-U/  EFF-U  PROC
  2 C1SYS1   *STDPool  30.00    2.00/  20.00  30.00/  30.00  2(+1)
  3 C1SYS2   *STDPool  30.00    1.00/  10.00  100.00/ 30.00  2(+1)

VM-ID      SCHED  ACT-I  VIRT  CPUS
  2 C1SYS1   TS     NO     00(RUN),01(RUN),02(OFF)
  3 C1SYS2   TS     NO     00(RUN),01(RUN),02(OFF)

...

/delete-vm vm-id=c1sys2 _____ (9)
VMS4429 VIRTUAL MACHINE (3,C1SYS2) REMOVED FROM VM GROUP 'GCUST1'
VMS4619 VIRTUAL MACHINE (3,C1SYS2) REMOVED FROM CPU POOL '*STDPool'
VMS4024 VIRTUAL MACHINE (3,C1SYS2) TERMINATED
/remove-vm-from-vm-group vm-id=c1sys1,vm-group-name=gcust1,
                          cpu-quota=30 _____ (10)
VMS4429 VIRTUAL MACHINE (2,C1SYS1) REMOVED FROM VM GROUP 'GCUST1'

```

- (8) The information output for the VM group shows the status now achieved: The two customer VMs `C1SYS1` and `C1SYS2` share the specified CPU capacity in a ratio of 2:1, i.e. 20% and 10% of the server's CPU capacity (planned values). The maximum CPU utilization of `C1SYS2` also corresponds to the specified CPU capacity because of the settings for the VM group.
- (9) After a certain time the customer no longer requires the additionally created VM `C1SYS2`. The VM2000 administrator terminates the VM.
- (10) The VM `C1SYS1` is removed from the VM group. It (again) becomes an autonomous VM with a CPU quota of 30 as the customer still wishes to use the newly defined CPU capacity. The member CPU quota is relinquished.


```
/show-vm-group vm-group-name=gcust1,information=*members _____ (11)
```

```
VM-GROUP CPU-POOL VM-ID          CPU-Q / EFF-Q  MAX-U / EFF-U
GCUST1   *STDPOOL  NONE
```

```
/delete-vm-group vm-group-name=gcust1 _____ (12)
```

```
VMS4629 VM GROUP 'GCUST1' REMOVED FROM CPU POOL '*STDPOOL'
VMS4405 VM GROUP 'GCUST1' DELETED
```

```
/show-vm-resources information=*std _____ (13)
```

```
          CPU-POOL(P)
VM-ID     VM-GROUP(G) CPU-Q   MAX-U   PROC   MEM-SIZE  MAX-IO
  1 MONITOR *STDPOOL(P) 30.00  100.00  2(+1)   128 MB (100)
  2 C1SYS1  *STDPOOL(P) 30.00   30.00  2(+1)   128 MB (100)
```

```
VM-ID     STATE           GUEST  VERSION ADMIN PRIV
  1 MONITOR  RUNNING           BS2-OSD V09.0A YES  AS
  2 C1SYS1  RUNNING           BS2-OSD V09.0A NO   AS
F R E E  VM:13          LARGEST-DOMAIN: 3814 MB
```

- (11) The VM group GCUST1 thus no longer has any members and can be deleted.
- (12) The VM group GCUST1 is deleted.
- (13) The information output shows that the required status has been achieved.

5.4 Operating the whole VM2000 system

The VM2000 system as a whole is operated by the VM2000 administrator. He/she is the privileged administrator who is authorized to enter all VM2000 commands.

The same person is also the VM administrator for the monitor VM.

The VM2000 administrator always has priority over the VM administrator. For example, the VM2000 administrator can intervene in every VM.

The VM2000 administrator operates the whole VM2000 system via the monitor VM in the ADMIN dialog.

The whole VM2000 system can also be operated from BS2000 consoles in the monitor system, e.g. if BCAM is not active in the monitor system.



The authorization code “9” is required to operate the VM2000 system as a whole via a BS2000 console of the monitor system.

All VM2000 commands have the authorization code “9”.

All VM2000 messages have the routing code “9”.

Only VM2000 commands can be entered and only VM2000 messages can be received on a BS2000 console that **only** has the authorization code “9”. This means that a BS2000 console of this type has the same authorization as the ADMIN dialog of the VM2000 administrator.

The VM2000 administrator can use:

- privileged VM2000 commands (VM2000 administrator commands)
- privileged functions in the VM administrator commands

The standard range of commands/functions of the VM2000 commands can be restricted for the VM2000 administrator or extended for the VM administrator (see [page 192](#)).

See also: “[Logging](#)” on [page 93](#) and [page 116](#).

5.4.1 Types of command

VM2000 commands can be split up into two groups:

- VM2000 administrator commands e.g. /CREATE-VM
- VM administrator commands e.g. /START-VM)

VM2000 administrator commands


- are normally intended for the exclusive use of the VM2000 administrator (identical to the monitor VM administrator)
- can be entered in the ADMIN dialog with the monitor VM or (e.g. if BCAM is not active in the monitor system) at BS2000 consoles in the monitor system
- require the OPERATING privilege or the VM2000-ADMINISTRATION privilege in the ADMIN dialog via a user task
- are used to operate the whole VM2000 system.



VM administrator commands

- are normally intended for use by the VM administrator
- require the OPERATING privilege or the VIRTUAL-MACHINE-ADMINISTRATION privilege in the ADMIN dialog via a user task
- can only be input by the VM administrator in the ADMIN dialog for his/her VM
- are used to operate a virtual machine
- can be used by the VM2000 administrator for any other VMs
- can have operand values whose use is restricted to the VM2000 administrator

Exceptions arise when the command/function range is extended and reduced for the VM and VM2000 administrator (ADMIN-PRIVILEGES, see [page 192](#)).

5.4.2 Overview of commands for VM2000 operation

Operation group	Command	For VM2000 administrator only	Privileged functions	Restrictions
Initializing a VM	CREATE-VM	x		(4)
Managing devices for a VM	ADD-VM-DEVICES	x	(6)	
	REMOVE-VM-DEVICES		(1)	
	SWITCH-VM-DEVICES	x		
	MODIFY-VM-DEVICE-ATTRIBUTES	x		
	MODIFY-VM-DEVICE-USAGE	x	(6)	
Managing Assignment sets (see also page 288)	CREATE-VM-ASSIGNMENT-SET	x		
	DELETE-VM-ASSIGNMENT-SET	x		
	SHOW-VM-ASSIGNMENT-SET	x		
Starting administration or operation	BEGIN-VM-DIALOG			(5)
Starting the guest system on a VM	START-VM		(1)	(4)
Information services	SHOW-VM-ATTRIBUTES		(1) (2)	
	SHOW-VM-DEVICE-STATUS	x	(6)	
	SHOW-VM-RESOURCES	x	(6)	
	SHOW-VM-STATUS	x		
Resource management (real CPUs) ( global storage) (main memory)	MODIFY-VM-ATTRIBUTES	x	(6)	
	ATTACH-VM-RESOURCES	x		
	DETACH-VM-RESOURCES	x		
	ADD-VM-RESOURCES	x	(6)	
	REMOVE-VM-RESOURCES		(1)	
	EXTEND-VM-MEMORY	x		
	REDUCE-VM-MEMORY	x		
Command processing via VM2000 procedure	CALL-VM-PROCEDURE		(1)	(5), (7)
Suspending a VM	HOLD-VM		(1) (3)	(4)

Operation group	Command	For VM2000 administrator only	Privileged functions	Restrictions
Canceling a wait state	RESUME-VM		(1) (3)	(4)
 Moving a VM within main memory	MOVE-VM	x		(4)
Passing information to VM administrators	SEND-VM-MESSAGE			
Modifying parameters for \$VMCONS	MODIFY-VMCONS-PARAMETERS	x		
Terminating administration or operation	END-VM-DIALOG		(1)	
Aborting a guest system	RESET-VM		(1)	(4)
Shutdown in the guest system and of VM2000	SHUTDOWN-VM		(1) (8)	(4)
Terminating a VM	DELETE-VM		(1) (3)	(4)
 Operating VM groups (see also page 176)	CREATE-VM-GROUP	x		
	ADD-VM-TO-VM-GROUP	x		
	MODIFY-VM-ATTRIBUTES	x	(6)	
	REMOVE-VM-FROM-VM-GROUP	x		
	MODIFY-VM-GROUP-ATTRIBUTES	x		
	SHOW-VM-GROUP	x		
	DELETE-VM-GROUP	x		
Managing CPU pools (see also page 225)	CREATE-VM-CPU-POOL	x		
	 ASSIGN-VM-GROUP-TO-CPU-POOL	x		
	ASSIGN-VM-TO-CPU-POOL	x		
	SWITCH-VM-CPU	x		
	SHOW-VM-CPU-POOL	x		
	DELETE-VM-CPU-POOL	x		
Managing the VM2000 configuration (see also page 363)	SAVE-VM-CONFIGURATION	x		
	RESTORE-VM-CONFIGURATION	x		

- (1) The VM2000 administrator may also specify the VM-IDs of other VMs. The VM administrator is only allowed to specify the VM-ID of his/her own VM.
- (2) With `VM-ID=*VM2000/*BY-VM-GROUP/*BY-CPU-POOL/<name 1..8 with-wild(20)`, the VM2000 administrator can request information about several VMs.
- (3) The VM2000 administrator can manage all VMs by means of `VM-ID=*ALL`.
- (4) This command is rejected for the monitor VM.
- (5) This command cannot be entered from a BS2000 console of the monitor system.
- (6) This command can also be used by VM administrators with `ADMIN-PRIVILEGES=*MAXIMUM` (see [page 193](#)).
- (7) This command cannot be issued from a user task.
- (8) Here the VM2000 administrator may terminate all guest systems using `VM-ID=*VM-GUESTS`, and the overall VM2000 system using `VM-ID=*VM2000`.

5.4.3 Privileged functions

The VM administrator can use the VM administrator commands only on his/her VM. The VM2000 administrator can also use VM administrator commands on any VMs. The VM2000 administrator has the following individual privileges:

- **Addressing any VMs**
The VM2000 administrator may specify the VM-IDs of other VMs. The VM administrator is only allowed to access his/her own VM, which was assigned using `/BEGIN-VM-DIALOG`.
- **Managing all VMs**
With the aid of `/HOLD-VM`, `/RESUME-VM` and `/DELETE-VM`, the VM2000 administrator can control operations on all VMs (`VM-ID=*ALL`).
Using the `/SHUTDOWN-VM` command, the VM2000 administrator can manage all VMs with the exception of the monitor VM (`VM-ID=*VM-GUESTS`).
- **Specifying information for several VMs**
Using `/SHOW-VM-ATTRIBUTES`, the VM2000 administrator can output information for several VMs (`VM-ID=*VM2000/*BY-VM-GROUP/*BY-CPU-POOL/<name 1..8 with-wild(20)`).

5.4.4 Restrictions in VM2000 commands/functions

Restrictions affecting the monitor VM and the monitor system

Some VM2000 commands must not be used for the monitor VM or the monitor system. The following restrictions apply:

- The monitor VM **cannot**
 - be initialized (`/CREATE-VM`)
 - be started using `/START-VM`
 - be suspended (`/HOLD-VM`, `/RESUME-VM`)
 - be moved within main memory (`/MOVE-VM`)
 - be terminated (`/DELETE-VM`)

Monitor VM and monitor system are initialized and started during initialization of VM2000 operation and terminated with `/SHUTDOWN-VM VM-ID=*VM2000`, with the BS2000 shutdown of the monitor system or, on SQ servers, via the SQ manager.

- The monitor system **cannot**
 - be aborted (`/RESET-VM`)
 - be terminated as an individual guest system using `/SHUTDOWN-VM`
- Virtual consoles are **prohibited** for the monitor VM (`/ADD-VM-DEVICES . . . , TYPE=*VC`).
- The monitor VM cannot be monitored with MONJVs.
- Global storage (S servers) can only be assigned to the monitor VM in real mode.

Special considerations regarding input from a BS2000 console of the monitor system

- The commands `/BEGIN-VM-DIALOG` and `/CALL-VM-PROCEDURE` cannot be entered from a BS2000 console of the monitor system.
- VM2000 procedures can be called from BS2000 consoles of the monitor system by means of `/RUN`, in which case the commands `/REMARK` and `/STEP` are ignored. The procedure must not contain any continuation lines.
- In all commands entered from a BS2000 console of the monitor system, the operand `VM-ID=*CURRENT` is assumed as the selection for the monitor VM.

5.4.5 Extending and restricting the range of commands/functions

The standard division into VM2000 administrator commands and VM administrator commands can be modified in the following two ways:

- restricting the command or function range for the VM2000 administrator
- extending the command or function range for the VM administrator.

These modifications are controlled by the `ADMIN-PRIVILEGES` operand in the commands `/CREATE-VM` and `/MODIFY-VM-ATTRIBUTES`.

Restrictions for the VM2000 administrator

The command/function range for the VM2000 administrator can be restricted:

- S On S servers when initializing the monitor VM, with `/CREATE-VM ... ,ADMIN-PRIVILEGES=*MINIMUM`.
- in the current VM2000 session, with `/MODIFY-VM-ATTRIBUTES ... ,ADMIN-PRIVILEGES=*MINIMUM`

The following table shows the restrictions for the individual commands when `ADMIN-PRIVILEGES=*MINIMUM` is specified.

Operation group	Command	Restrictions
Starting the guest system on a VM	START-VM	(1)
Resource management	MODIFY-VM-ATTRIBUTES	(3)
Command processing via procedure call	CALL-VM-PROCEDURE	(2)
Suspending a VM	HOLD-VM	(1)
Canceling a wait state	RESUME-VM	(1)
Terminating ADMIN or VC dialog	END-VM-DIALOG	(2)
Output of VM2000 monitored data	SHOW-VM-STATUS	(2)

- (1) These commands are no longer usable by the VM2000 administrator.
- (2) The VM2000 administrator may use these commands only for the monitor VM.
- (3) This command can be used by the VM2000 administrator for the monitor VM with the exception of the `ADMIN-PRIVILEGES` operand. The VM2000 administrator can only use this command for another VM with the operand `ADMIN-PRIVILEGES=*MAXIMUM`.

All other commands remain unaffected. This enables VM2000 operation in which the VM2000 administrator is largely separated from the VMs.

For reasons of security it is **not** possible to revoke a restriction imposed by the operand `ADMIN-PRIVILEGES=*MINIMUM` during the current VM2000 session.

Extensions for the VM administrator

The command/function range can be extended for the VM administrator:

- when initializing a VM (except for the monitor VM), with `/CREATE-VM . . . ,ADMIN-PRIVILEGES=*MAXIMUM`
- after initializing a VM (except for the monitor VM), with `/MODIFY-VM-ATTRIBUTES . . . ,ADMIN-PRIVILEGES=*MAXIMUM`

The following table shows the extensions for the individual commands if `ADMIN-PRIVILEGES=*MAXIMUM` is specified.

Operation group	Command	Extensions
Device management for a VM	ADD-VM-DEVICES	(1)
	MODIFY-VM-DEVICE-USAGE	(1)
Displaying information	SHOW-VM-DEVICE-STATUS	(2)
	SHOW-VM-RESOURCES	(2)
Resource management	MODIFY-VM-ATTRIBUTES	(1)
	ADD-VM-RESOURCES	(1)

- (1) The VM administrator may also use these commands for his/her own VM, with all the attendant functions.
- (2) The VM administrator may also use these commands, with all the attendant functions.

All other commands remain unaffected. The VM administrator cannot access any other VMs. This enables VM2000 operation in which some of the management functions can be dealt with by the VM administrators (e.g. test runs).

The extensions are revoked with `/MODIFY-VM-ATTRIBUTES . . . ,ADMIN-PRIVILEGES=*STD` or with `/DELETE-VM`.

5.5 VM2000 in the guest systems

VM2000 in the monitor system and the other BS2000 guest systems

This section is also aimed at systems support in all BS2000 guest systems (even in the monitor system). It describes how (guest) systems support receives information on VM2000 and how it can use VM2000 functions for the guest system. In addition, the VM2000 functions are mentioned whose execution entails the interworking of VM2000 and the guest system.

A guest system under VM2000 is partitioned off from the other guest systems. It behaves in production operation like an operating system in native operation. The following options for information output and interaction are offered for a guest system (thus also for the monitor system):

For all server series

- The nonprivileged macro VMGINF provides global VM2000 information (see the “Executive Macros” manual [15]).
- Output of global VM2000 information in all guest systems (see [page 202](#) and the BS2000 command /SHOW-SYSTEM-INFORMATION in the “Commands” manual [12])
- Output of VM2000 information on devices and pubsets in the guest system as well (see [page 206](#) and the BS2000 command /SHOW-DEVICE-CONFIGURATION in the “Commands” manual [12])
- In the case of /SHUTDOWN-VM, VM2000 and the guest systems concerned interwork to execute shutdown processing (BS2000/OSD as of V7.0, see [page 79](#))
- Dynamic main memory reconfiguration is carried out by VM2000 together with the guest system (see [page 270](#)).
- Implicit assignment of (pubset) devices to the VM and implicit release of (pubset) devices of the VM by the operator in the guest system (see [page 288](#) and the description of the BS2000 reconfiguration commands mentioned there as of [page 197](#) and in the “Commands” manual [12]).
- For devices that are attached in the guest system and that are possibly still in use, explicit device assignment can be converted to an implicit device assignment. The device then remains with the VM until the guest system detaches the device (see “Explicit removal of devices” on [page 287](#))
- The BS2000 utility routine IORM can be used in all guest systems. It operates on a cross-guest-system basis and is administered in the monitor system. The IO limit of the IORM function IOLVM is set on S servers by means of VM2000 commands (See [page 359](#) and the “Utility Routines” manual [3].)

- In the case of measurements with the openSM2 measurement monitor, measurement information on the VM of the guest system is output in the VM2000 report and in the SM2R1 reports 98 and 99 (see [page 392](#) and the “openSM2” manual [8])
- The guest systems are informed about the failure of the monitor system and about the renewed readiness of the system for operation (messages NRTV001 – NRTV003, see [page 402](#)).

For S servers

- In the case of /MOVE-VM, VM2000 and the guest systems concerned interwork to provide protection in the MSCF network (BS2000/OSD as of V7.0, see [page 166](#)).
- Using the BS2000 command /MODIFY-IO-UNIT you can specify in the guest systems a different base or alias device that is to be preferred under VM2000 for I/Os of this guest system (see [page 347](#) and [page 201](#) and in the “Commands” manual [12]).
- SLEDFILES with the dump object VM2000 can be evaluated with DAMP (see [page 412](#) and the “Diagnostics Handbook” manual [4]).
- In the case of guest system SLEDs, the diagnostic data of the VM2000 hypervisor is contained in the SLEDFILE (see [page 398](#) and the “Introductory Guide to Systems Support” manual [2])

VM2000 in the monitor system

The monitor system has a special role. The following options for information output and controlling the guest system operation are additionally offered for the monitor system.

For all server series

- The security administrator in a BS2000 guest system can use the `VIRTUAL-MACHINE-ADMINISTRATION` and `VM2000-ADMINISTRATION` privileges to permit the operation of his/her own VM or of VM2000 for a user task using VM2000 commands (see [page 91](#) and the “Introductory Guide to Systems Support” [2]).
- In measurements using the openSM2 measurement monitor, measurement information on all VMs is output in the VM2000 report and in SM2R1 reports 98 and 99 (see [page 392](#) and the “openSM2” manual [8]).

For S servers

- (global) reconfiguration of hardware units and paths in all the active guest systems by the operator in the monitor system (see [page 262](#) and the description of the BS2000 reconfiguration commands mentioned there as of [page 197](#) and in the “Commands” manual [12])
- Interplay between VM2000 and the system support in the monitor system during the virtualization of the global storage unit (see [page 278](#) and the “Introductory Guide to Systems Support” manual [2])
- Interplay between VM2000 and the operating in the monitor system during dynamic I/O configuration modification of channel devices (see [page 361](#) and the “Introductory Guide to Systems Support” manual [2])

ATTACH-DEVICE

Attach hardware units

This command allows the operator to make one or more hardware units available, i.e. the operating system is permitted to use these units for I/O operations.

Format (VM2000-specific operands)

ATTACH-DEVICE
<pre> UNIT = ... / *CHANNEL(...) / *CONTROLLER(...) / *CHANNEL-RANGE(...) / *PUBSET-DEVICES(...) / *CHANNEL(...) CHANNEL-PATH-ID = list-poss(16): <x-text 2..2> ,SCOPE = *OWN-SYSTEM-ONLY / *VM2000-GLOBAL *CONTROLLER(...) CONTROLLER-UNIT = list-poss(16): <alphanum-name 2..2> / <x-text 4..4> ,SCOPE = *OWN-SYSTEM-ONLY / *VM2000-GLOBAL *CHANNEL-RANGE(...) FROM = <x-text 2..2> ,TO = <x-text 2..2> ,SCOPE = *OWN-SYSTEM-ONLY / *VM2000-GLOBAL *PUBSET-DEVICES(...) PUBSET = list-poss(255): <cat-id 1..4> / *BY-PUBRES-DEVICE(...) *BY-PUBRES-DEVICE(...) UNIT = list-poss(255): <alphanum-name 2..2> / <x-text 4..4> ... </pre>

Description of the operands (VM2000-specific operands)

UNIT = *CHANNEL(...) / *CONTROLLER(...) / *CHANNEL-RANGE(...) / *PUBSET-DEVICES(...)

Specifies the channels or controllers or pubsets that are to be attached. The devices of a pubset can also be controlled as a unit (with analogous syntax `*PUBSET-DEVICES(...)`) in the VM2000 commands.

SCOPE = (nur für *CHANNEL, *CONTROLLER, *CHANNEL-RANGE)

Specifies how the command is to be executed under VM2000.

SCOPE = *OWN-SYSTEM-ONLY

The command is only executed in the local system.

SCOPE = *VM2000-GLOBAL

If entered on the monitor system (S-Server), the command is executed on all guest systems.

If entered in another guest system or in the monitor system of an SQ server, the command is rejected with message NKR0178.

DETACH-DEVICE**Detach hardware units**

This command allows the operator to detach one or more hardware units from the system and thus prevent the operating system from using these units.

Format (VM2000-specific operands)

DETACH-DEVICE
<pre> UNIT = ... / *CHANNEL(...) / *CONTROLLER(...) / *CHANNEL-RANGE(...) / *PUBSET-DEVICES(...) / *CHANNEL(...) CHANNEL-PATH-ID = list-poss(16): <x-text 2..2> ,SCOPE = *OWN-SYSTEM-ONLY / *VM2000-GLOBAL *CONTROLLER(...) CONTROLLER-UNIT = list-poss(16):<alphanum-name 2..2> / <x-text 4..4> ,SCOPE = *OWN-SYSTEM-ONLY / *VM2000-GLOBAL *CHANNEL-RANGE(...) FROM = <x-text 2..2> ,TO = <x-text 2..2> ,SCOPE = *OWN-SYSTEM-ONLY / *VM2000-GLOBAL *PUBSET-DEVICES(...) PUBSET = list-poss(255): <cat-id 1..4> / *BY-PUBRES-DEVICE(...) *BY-PUBRES-DEVICE(...) UNIT = list-poss(255): <alphanum-name 2..2> / <x-text 4..4> ... ,FORCE = *STD / *YES / *NO(...) / *UNCONDITIONAL-OFFLINE </pre>

Description of the operands (VM2000-specific operands)**UNIT = *CHANNEL(...)/ *CONTROLLER(...)/ *CHANNEL-RANGE(...)/ *PUBSET-DEVICES(...)**

Specifies the channels or controllers or pubsets that are to be detached. The devices of a pubset can also be controlled as a unit (with analogous syntax *PUBSET-DEVICES(. . .)) in the VM2000 commands.

SCOPE = (nur für *CHANNEL, *CONTROLLER, *CHANNEL-RANGE)

Specifies how the command is to be executed under VM2000.

SCOPE = *OWN-SYSTEM-ONLY

The command is only executed in the local system.

SCOPE = *VM2000-GLOBAL

If entered on the monitor system (S-Server), the command is executed on all guest systems.

If entered in another guest system or in the monitor system of an SQ server, the command is rejected with message NKR0178.

The command is rejected for hardware units if one of the guest systems is protected against global detachment (system parameter VMGIORAL=NO) and if the hardware unit that is to be detached is not yet in the DETACHED state (unless FORCE=*UNCONDITIONAL-OFFLINE has been specified for channels).

. . .

FORCE =

Specifies the mode of execution.

FORCE = *UNCONDITIONAL-OFFLINE

Operand value for S servers only

The reconfiguration request is to be executed regardless. In contrast to FORCE=*YES, the hardware action associated with the request must always be carried out. This setting is only permissible when channels are detached (CHANNEL or CHANNEL-RANGE).

The protection of one of the guest systems against global detachment (system parameter VMGIORAL=NO) is ignored when this operand is specified.

INCLUDE-DEVICE-CONNECTION**Add logical connections**

See /REMOVE-DEVICE-CONNECTION.

REMOVE-DEVICE-CONNECTION

Remove logical connections

SQ Global reconfiguration of hardware units is not available on SQ servers. When VM2000-specific operands for global configuration are specified in these BS2000 commands, the commands are rejected with the message NKR0178.

The operator can use these commands to add or remove logical connections between the hardware units (CHN, CTL, DVC) to or from system, i.e. the operating system is permitted to use these connections or prevented from using them.

Format (VM2000-specific operands)

INCLUDE-DEVICE-CONNECTION / REMOVE-DEVICE-CONNECTION
FROM = *CHANNEL(...) / *CONTROLLER(...) / list-poss(8): <alphanum-name 2..2> / <x-text 4..4>
...
,SCOPE = *OWN-SYSTEM-ONLY / *VM2000-GLOBAL
...

Description of the operands (VM2000-specific operands)

SCOPE =

Specifies how the command is to be executed under VM2000.

SCOPE = *OWN-SYSTEM-ONLY

The command is only executed in the local system.

SCOPE = *VM2000-GLOBAL

If entered on the monitor system (S-Server), the command is executed on all guest systems.

If entered in another guest system or in the monitor system of an SQ server, the command is rejected with message NKR0178.

MODIFY-IO-UNIT

Modify the configuration properties of the input/output unit

SQ PAV devices (channel) are not available on SQ servers, where the inputs/outputs are automatically parallelized by X2000. The VM2000-specific operands for prioritizing PAV devices in this BS2000 command consequently have no effect on SQ servers.

This command allows systems support to modify the configuration properties of controllers or devices dynamically.

Format (VM2000-specific operands)

```

MODIFY-IO-UNIT

UNIT = *CONTROLLER(...) / *DEVICE(...)
...
*DEVICE(...)
    NAME = <alphanum-name 2..2> / <x-text 4..4> / *RANGE(...)
    ,STATE = *UNCHANGED / *PREFERRED-PATH(...) / *PAV-PREFERRED-DEVICE(...)
        *PAV-PREFERRED-DEVICE(...)
            ,PAV-PREFERRED-DEVICE = *UNCHANGED / *NO / *YES
...

```

Description of the operands (VM2000-specific operands)

UNIT = *DEVICE(...)

The configuration properties of a device are to be modified.

STATE = *PAV-PREFERRED-DEVICE(...)

Specifies whether the base or alias device should be given preference under VM2000 for I/Os.

PAV-PREFERRED-DEVICE = *UNCHANGED

The current setting remains unchanged.

PAV-PREFERRED-DEVICE = *YES

The base or alias device is preferred under VM2000 for I/Os.

PAV-PREFERRED-DEVICE = *NO

The preference is canceled.

SHOW-SYSTEM-INFORMATION

Display information on the current system

The user can use `/SHOW-SYSTEM-INFORMATION` to obtain data on the current system run.

You can obtain the following system information on VM2000 operation:

- information on the current system configuration (SYSID, home pubset, host name, VM identification, system name and origin of the system parameters at system startup)
- the VM2000 version
- information on the monitor system (BS2000/OSD version and host name)

The command supports structured output to S variables.

Format (VM2000-specific operands)

```
SHOW-SYSTEM-INFORMATION
```

```
INFORMATION = *ALL / ... / *HSI-ATTRIBUTES / *SYSTEM-CONFIGURATION /
              *VM2000-VERSION / *VM2000-MONITOR-SYSTEM / ...
,OUTPUT = *SYSOUT / *SYSLST
```

Description of the operands (VM2000-specific operands)

INFORMATION =

Specifies which system information is to be output.

...

INFORMATION = *HSI-ATTRIBUTES

Provides information on the hardware-software interface (HSI) in the form:

```
HSI-ATT:      TYPE                = IX / KM / UD
              ASF                  = NO / YES
              OPERATION-MODE      = NATIVE / VM2000
```

Meaning of the output values

OPERATION-MODE Provides information on the use of VM2000:

=NATIVE The operating system is not running as a guest system under VM2000.

=VM2000 The operating system is running as a guest system under VM2000.

INFORMATION = *SYSTEM-CONFIGURATION

Provides information on the configuration of the BS2000 system in the form:

```
SYSTEM-CONF:  SYSID           = <sys id>
               HOME-PUBSET    = <cat-id>
               HOST-NAME       = <host-name>
               VM-INDEX        = <vm-index>
               VM-NAME         = <vm-name>
               SYSTEM-NAME     = <sys-name>
               SYSPAR-BS2-SEL  = *STD / *BY-VM-NAME / *BY-SYSTEM-NAME / ...
```

Meaning of the output values

<vm-index> In VM2000 operation, the VM index in the format <integer 1..99>;
in native operation *NONE.

<vm-name> In VM2000 operation, the VM name in the format <name 1..8>; in
native operation *NONE.

*BY-VM-NAME Selection of the BS2000 parameter file SYSPAR.BS2.vvv.<vm-name>;
vvv corresponds to the BS2000 version without a period; <vm-name>
corresponds to VM name

*BY-SYSTEM-NAME Selection of the BS2000 parameter file SYSPAR.BS2.vvv.<sys-
name>; vvv corresponds to the BS2000 version without a period;
<sys-name> corresponds to system name

INFORMATION = *VM2000-VERSION

Provides information on the VM2000 version in VM2000 operation:

```
VM2000-VERSION        = <vm2000-version>
```

Meaning of the output values

<vm2000-version> Version in DOD format e.g. V10.0A

INFORMATION = *VM2000-MONITOR-SYSTEM

Provides information on the monitor system in VM2000 operation:

```
VM2000-MONITOR-  OSD-BC-VERSION  = <osd-bc-version>
SYSTEM:          HOST-NAME      = <host-name>
```

Meaning of the output values

<osd-bc-version> Version in DOD format, e.g. V09.0A0000

<host-name> BCAM name of the monitor system, e.g. D016ZE10

Example (in the monitor VM)

```

/show-system-information information=*all _____ (1)
%CONFIGURATION = 7.500- S210-20
%CPU-ID-LIST :   ADR   0 = 1002000122600000
%              ADR   1 = 1012000122600000
%              ADR   2 = 1022000122600000
%HSI-ATT :      TYPE = IX _____ (2)
%              ASF = YES
%              OPERATION-MODE = VM2000
%MEMORY-SIZE = 320 MB
%MINIMAL-MEMORY-SIZE = 256 MB
%BS2000-ID :    NAME = L15BXS
%              VERSION = V17.0A0000
%              OSD-BC-VERSION = V08.0A0000
%              CREATED = 2011-04-11,10:21:30
%IOCONF-ID :   NAME = S2100001
%              VERSION = V18.0A00
%              CREATED = 2012-01-11,14:59:09
%              FORMAT = IORSF01
%IPL-TIME = 2012-01-19,21:07:36
%SYSTEM-CONF : SYSID = 180 _____ (3)
%              HOME-PUBSET = MONR
%              HOST-NAME = D017ZE03
%              VM-INDEX = 1
%              VM-NAME = MONITOR
%              SYSTEM-NAME = *NONE
%              SYSPAR-BS2-SEL = *STD
%VM2000-VERSION = V10.0A _____ (4)
%VM2000-MONITOR- OSD-BC-VERSION = V08.0A0000 _____ (5)
% SYSTEM:      HOST-NAME = D017ZE03
%SYSTEM-TIME-  ZONE = +01:00
% PARAMETER:   SEASON = W
%              SEASON-DIFFERENCE = 01:00
%              PREV-CHANGE-DATE = 2011-10-30,03:00:00
%              NEXT-CHANGE-DATE = 2012-03-25,02:00:00
%              SYNCHRONIZATION = SERVER-CONN-EXT-REF

```

- (1) All the information of the command is to be displayed
- (2) Corresponds to the output for INFORMATION=*HSI-ATTRIBUTES
- (3) Corresponds to the output for INFORMATION=*SYSTEM-CONFIGURATION
- (4) Corresponds to the output for INFORMATION=*VM2000-VERSION
- (5) Corresponds to the output for INFORMATION=*VM2000-MONITOR-SYSTEM

Output in S variables

The command provides information on the current system that is running.

INFORMATION=*ALL outputs all the variables.

Notation of command	Abbreviated form used in table
...	...
INFORMATION = *HSI-ATTRIBUTES	3
INFORMATION = *SYSTEM-CONFIGURATION	9
INFORMATION = *VM2000-VERSION	10
INFORMATION = *VM2000-MONITOR-SYSTEM	11

Output information	Name of the S variable	T	Contents	Condition
...				
Operational mode of the operating system	var(*LIST).HSI-ATTR.OPERATION-MODE	S	NATIVE VM2000	3
VM index (in VM2000 operation)	var(*LIST).SYS-CONF.VM-INDEX	I	<integer 1..99>	9
VM name (in VM2000 operation)	var(*LIST).SYS-CONF.VM-NAME	S	<c-string 1..8>	9
VM2000 version in VM2000 operation, otherwise *NONE	var(*LIST).VM2000-VERSION	S	<c-string 6> *NONE	10
BCAM name of the monitor system (in VM2000 operation), otherwise *NONE	var(*LIST).VM-MONITOR.HOST-NAME	S	<c-string 8> *NONE	11
BS2000/OSD-BC version of the monitor system (in VM2000 operation), otherwise *NONE	var(*LIST).VM-MONITOR.OSD-BC-VERSION	S	<c-string 10> *NONE	11

SHOW-DEVICE-CONFIGURATION

Display system configuration and availability

`/SHOW-DEVICE-CONFIGURATION` provides information on the system configuration and the availability of the hardware units (refer also to the commands manual [12] for the meaning of the output columns of the device management `SHOW` commands).

The command supports structured output to S variables.

Format (VM2000-specific operands)

SHOW-DEVICE-CONFIGURATION	(VM2000 specific)
<pre> UNIT = *STD / *SELECT(...) / *PUBSET-DEVICES(...) / ... *SELECT(...) CLASS = *ALL / ... / *DEVICE(...) *DEVICE(...) TYPE = *ALL / <device> ,VM-ASSIGNMENT = *ANY / *NO / *YES / *BY-GUEST-POSSIBLE ,ATTRIBUTE = *PUBSET-DEVICES(...) PUBSET = <cat-id 1..4> / *BY-PUBRES-DEVICE(...) *BY-PUBRES-DEVICE(...) UNIT = <alphanum-name 2..2> / <x-text 4..4> ,INFORMATION = *STD / ... / *VM2000 </pre>	

Description of the operands (VM2000-specific operands)

UNIT = ***SELECT(...)** / ***PUBSET-DEVICES(...)**

Specifies a subset of the hardware units known to the system for which information is output. The hardware units are selected using device class and state or using a pubset. The devices of a pubset can also be controlled as a unit (with the same syntax `*PUBSET-DEVICES(...)`) in the VM2000 commands.



In order to obtain information on the pubset devices (`*PUBSET-DEVICES` operand), the guest system attempts, if necessary, to implicitly assign and attach the pubres device (and the volres devices) to the VM for the duration of command processing.

CLASS = *DEVICE(...)

Information on end devices is displayed.

TYPE = *ALL / <device>

Information is output on end devices of the specified device type. *ALL is the default. In other words, all the device types are displayed by default. When a device type is specified explicitly, only device or volume types that are known in the system are accepted. The possible device and volume types are displayed in the dialog with `DEVICE-TYPE=?`.

VM-ASSIGNMENT = *ANY / *NO / *YES / *BY-GUEST-POSSIBLE

Information is displayed on end devices that have the specified status. *ANY is the default. In other words, output is not dependent on this status by default.

VM-ASSIGNMENT = *NO

Information is displayed on end devices not assigned to own VM.

VM-ASSIGNMENT = *YES

Information is displayed on end devices assigned to own VM.

VM-ASSIGNMENT = *BY-GUEST-POSSIBLE

Information is displayed on end devices that are not assigned to their own VM but that can be temporarily assigned implicitly (using `/ATTACH-DEVICE`, for example).

...

INFORMATION = *VM2000

Information on possible assignment to own VM and on usage by own VM.

Output format:

```
MNEM UN-CLASS UN-TYPE CONF-STATE VM-ASSIGN VM-USAGE
```

Example

```

/show-device-configuration unit=(d2,d3,d4,d5,d6,d7,d8,p2,p3,m4),
                               information=*vm2000 _____ (1)
MNEM UN-CLASS UN-TYPE  CONF-STATE  VM-ASSIGN  VM-USAGE _____ (2)
D2   DVC      D3435   ATTACHED   BY-GUEST   SH(I)
D3   DVC      STDDISK  DET(EX)    BY-VM-ADM  SH(D)
D4   DVC      D3435   DET(EX)    BY-GUEST-POSS SH(D)
D5   DVC      STDDISK  DET(EX)    BY-GUEST-POSS NONE
D6   DVC      STDDISK  DET(EX)    ELSEWHERE   EXCL
D7   DVC      D3435   ATTACHED   BY-VM-ADM  EXCL
D8   DVC      D3435   ATTACHED   BY-VM-ADM  EXCL
P2   DVC      HNC     ATTACHED   BY-GUEST   EXCL
P3   DVC      HNC     ATTACHED   BY-GUEST   EXCL
M4   DVC      LTO-U3  DET(EX)    BY-GUEST-POSS EXCL

```

- (1) VM2000 information is requested for the devices specified
- (2) See [page 209](#) for the meaning of the output columns VM-ASSIGN and VM-USAGE

S variables

Output information	Name of the S variable	T	Contents	Condition
Attachment option for a VM	var(*LIST).VM-ASS	S	BY-GUEST BY-VM-ADM VM-INHIBIT DEV-INHIBIT ELSEWHERE BY-GUEST-POSSIBLE	INF=*VM2000
Use by a VM	var(*LIST).VM-USAGE	S	NONE EXCL SH(D) SH(I)	INF=*VM2000

Meaning of the output columns with INFORMATION=*VM2000

Keyword	Meaning
VM-ASSIGN	<p>Displays the device assignment or assignment options for a VM:</p> <p>BY-GUEST The device is implicitly assigned to its own VM.</p> <p>BY-VM-ADM The device is explicitly assigned to its own VM.</p> <p>BY-GUEST-POSS The device is not assigned to its own VM, and implicit assignment is possible.</p> <p>In the following cases, the device is not assigned to its own VM and implicit assignment is not possible:</p> <p>VM-INHIBIT The VM is never authorized for implicit assignment.</p> <p>DEV-INHIBIT The VM is not authorized to assign this device implicitly Exception for VM-INHIBIT / DEV-INHIBIT: Snap units (Snapsets) can nevertheless be implicitly assigned to a VM with the AUTO-SNAP-ASSIGNMENT privilege (see page 146).</p> <p>ELSEWHERE The device is assigned exclusively to another VM.</p>
VM-USAGE	<p>Displays the usage mode in which the device is assigned (depends on VM-ASSIGN). <i>If VM-ASSIGN = BY-GUEST / BY-VM-ADM:</i></p> <p>EXCL The device is assigned exclusively to its own VM.</p> <p>SH(D) The device is assigned to its own VM for shared usage with direct I/O processing (only one VM sharer).</p> <p>SH(I) The device is assigned to its own VM for shared usage with indirect I/O proc.</p> <p><i>If VM-ASSIGN = BY-GUEST-POSS:</i></p> <p>EXCL The device is assigned exclusively to another VM. It is still possible to assign the device due to the KEEP state of the device (MTC).</p> <p>SH(D) The device is assigned to another VM for shared usage with direct I/O processing (only one VM sharer).</p> <p>SH(I) The device is assigned to another VM for shared usage with indirect I/O processing.</p> <p>NONE The device is not assigned.</p> <p><i>If VM-ASSIGN = VM-INHIBIT / DEV-INHIBIT:</i></p> <p>EXCL The device is assigned exclusively to another VM.</p> <p>SH(D) The device is assigned to another VM for shared usage with direct I/O processing (only one VM sharer).</p> <p>SH(I) The device is assigned to another VM for shared usage with indirect I/O proc.</p> <p>NONE The device is not assigned.</p> <p><i>If VM-ASSIGN = ELSEWHERE:</i></p> <p>EXCL The device is assigned exclusively to another VM.</p>

6 VM2000 management

The management of VM2000 covers resource management in VM2000 operation and other organizational tasks.

Resource management in VM2000 operation includes

- CPU management and reconfiguration
- reconfiguration of hardware complexes, hardware units and paths
- management and reconfiguration of storage media
- management of the (individual) devices and pubset devices
- Backing up and restoring the VM2000 configuration

Other organizational tasks are

- password protection
- accounting procedure
- hardware error logging
- creation of monitored data.

See also: [“Logging” on page 93](#) and [page 116](#).

6.1 CPU management

CPU management under VM2000 includes:

- [Managing and reconfiguring real and virtual CPUs](#)
- [Managing CPU pools](#)
- [Planning distribution of the CPU capacity to the VMs](#)
- [Scheduling procedures](#)

A general example for CPU management is provided in the [section “Example of CPU management” on page 253](#).

6.1.1 Managing and reconfiguring real and virtual CPUs

These management tasks comprise:

- Managing the real CPUs which are available for VM2000 operation
- Managing the virtual CPUs of a VM
- Reconfiguration of real and virtual CPUs
- Managing the spare CPUs of the server
- Managing the extra CPUs of the server



Note on the representation of CPUs

A (real or virtual) CPU is always represented by its CPU identifier (CPU ID), which is a two-digit hexadecimal number. This applies to both command input (SDF format: `<x-text 1..2>`) and outputs.

The VM2000 commands, the outputs of VM2000 information commands and messages in which the CPU number is inserted are affected by this.

6.1.1.1 Real CPUs (processors)

The real CPUs of a BS2000/OSD server are subdivided into:

- CPUs which are available in VM2000 operation
 - ☐ On S servers all real CPUs are available for VM2000 operation.
 - ☐ For SQ servers, the number depends on the server model and the licenses, see also [page 34](#).
 - Real normal CPUs (type of CPU: NORMAL)
 - Real extra CPUs (type of CPU: EXTRA, license required)
 - Real hot spare CPUs (type of CPU: SPARE, S servers)

VM2000 manages real normal and extra CPUs in the CPU pools, see [section “Managing CPU pools” on page 225](#). The use of these CPUs under VM2000 is described in the following sections.

Note on the terms used

In order to distinguish them clearly from real spare CPUs and extra real CPUs, the real CPUs described here, where necessary, are referred to as **real normal CPUs**.

- ☐ When the monitor system starts up, **all** available and operable real (normal) CPUs of S servers are added to the standard CPU pool and started.

If hot spare CPUs or extra CPUs are available, these are placed in a ready status at startup. They are, however, initially only kept in the ready status. Extra CPUs are initially added to the standard CPU pool. Spare CPUs do not belong to any CPU pool.

- CPUs which are not available in VM2000 operation
 - Real CPUs for the XenVMs (including domain 0) (type of CPU: FOREIGN, SQ servers)
 - Real CPUs which are unused in this session, e.g. because no license exists or real cold spare CPUs (type of CPU: UNUSED, SQ servers)

Real CPUs are in one of the following statuses:

- ATT** The CPU is attached. This status is attained after
- successful startup
 - `successful /ATTACH-VM-RESOURCES`.
- DET** The CPU is detached. This status is attained after
- startup for a hot spare CPU or extra CPU
 - `successful /DETACH-VM-RESOURCES`.
- WFM** The CPU is waiting for maintenance after an error (Wait For Maintenance).
[S] It has been replaced by a hot spare CPU (if available) and can only be attached again after maintenance.
- ERR** The CPU has an error (S servers).
[S] It has not been replaced by a hot spare CPU and can be attached again immediately.
- OFF** The CPU is not available (offline).

The VM2000 administrator uses the following VM2000 commands to obtain information on the status of the real CPUs and the CPU pools to which they belong:

- `/SHOW-VM-CPU-POOL INFORMATION=*CPU`
 - `/SHOW-VM-RESOURCES INFORMATION=*CPU, VM-IDENTIFICATION=*VM2000`
 - `/SHOW-VM-RESOURCES INFORMATION=*CONFIGURATION`
- [SQ]** On SQ servers these commands display **all** real CPUs of the server unit, i.e. also CPUs of type `FOREIGN` or `UNUSED`. CPUs of the types `FOREIGN` and `UNUSED` are always in the `ATT` status.

6.1.1.2 Virtual CPUs

The number of CPUs for a VM (virtual CPUs) is defined by specifying a **multiprocessor level** when a VM is created (in /CREATE-VM by means of the PROCESSOR operand). A VM can be set up with the multiprocessor levels 1 through 32 (hardware dependent), in other words with a maximum of 32 virtual CPUs (CPU 00 through CPU 1F), see [page 143](#).

The multiprocessor level of a VM must be less than or equal to the number of real normal CPUs which can be available for VM2000 operation.

Exception: see the note on the PROCESSOR=*EXTRA-AND-NORMAL operand on [page 224](#).

Once a VM has been created, it is no longer possible to modify its multiprocessor level.



Depending on the expected load on the guest system, the multiprocessor level of a VM should be set as low as possible.

The multiprocessor level selected for a VM should also be less than or equal to the number of attached real CPUs in the CPU pool to which the VM is assigned (see [page 225](#)).

However, a guest system can also run if its multiprocessor level exceeds the number of attached real CPUs (overdimensioned VM).

Example

Biprocessor VM which is assigned to a CPU pool with two real CPUs; one of these real CPUs is detached. The resulting performance loss, e.g. through CPU locks of the virtual CPUs, must be taken into consideration.

Note on the terms used

In order to distinguish them clearly from virtual spare CPUs, the virtual CPUs described here, where necessary, are referred to as **virtual normal CPUs**.

For the monitor VM, the multiprocessor level is set when VM2000 is installed (see [page 62](#)).



If hot spare CPUs are available on S servers, each VM is assigned not only virtual normal CPUs but also virtual spare CPUs, see [page 220](#).

Virtual CPUs are assigned one of the following statuses:

- RUN** The CPU can run. This status is attained after
- successful startup of the guest system
 - successful `/ATTACH-DEVICE`
 - the CPU pool of an active VM has been switched with automatic attachment of virtual CPUs
(`/ASSIGN-VM-TO-CPU-POOL . . . ,ATTACHED-VM-CPUS=*ADJUST-NUMBER`)
 - the CPU pool has been extended with automatic attachment of virtual CPUs
(`/SWITCH-VM-CPU . . . ,TARGET-CPU-POOL=*ADJUST-NUMBER`)
- IDLE** The CPU is in the interruptible wait state.
- INIT**
- [SQ] The CPU is initialized (after configuration of the VM and up to automatic attachment in the case of startup in the BS2000 guest system).
- WAIT** The CPU has been stopped by VM2000 (VM in IN HOLD(WAIT) status).
- HALT**
- [SQ] The CPU has been halted by X2000 (short-term transitional status or error).
- STOP** The CPU has been stopped (stop by the hardware). This status is attained after
- successful `/DETACH-DEVICE`
(VM not overdimensioned, see [page 228](#))
 - the CPU pool of an active VM has been switched without automatic attachment of virtual CPUs (for CPUs in BLOCK status)
(`/ASSIGN-VM-TO-CPU-POOL . . . ,ATTACHED-VM-CPUS=*CHECK-NUMBER`)
 - the CPU pool has been extended without automatic attachment of virtual CPUs (for CPUs in BLOCK status)
(`/SWITCH-VM-CPU . . . ,TARGET-CPU-POOL=*NONE`)
- [S] – an error (machine check error)
- In these cases the CPU can be attached again during ongoing operation.
- BLOCK** The CPU is “blocked” by VM2000. This status is attained after the following actions:
- guest system startup (VM overdimensioned, see [page 228](#))
 - successful `/DETACH-DEVICE` (VM previously overdimensioned)
 - the CPU pool of an active VM has been switched with automatic detachment of virtual CPUs
(`/ASSIGN-VM-TO-CPU-POOL . . . ,ATTACHED-VM-CPUS=*ADJUST-NUMBER`)
 - the CPU pool has been reduced with automatic detachment of virtual CPUs
(`/SWITCH-VM-CPU . . . ,SOURCE-CPU-POOL=*ADJUST-NUMBER`)
 - the CPU pool of an active VM has been switched with preparatory detachment of virtual CPUs
 - the CPU pool of an active VM has been reduced with preparatory detachment of virtual CPUs

In these cases the CPU in the guest system cannot be attached again.

- SLEEP** The hot spare CPU is ready but sleeping.
- Ⓢ This VM currently contains only one attached virtual normal CPU.
- This status can also occur temporarily during a CPU reconfiguration.
- OFF** The CPU is not ready (offline). This status is attained
- Ⓢ – for a hot spare CPU if there are several attached virtual normal CPUs in this VM
 - Ⓢ – after a CPU error (MCK, MFA) if the CPU can only be used again after the next /START-VM for the guest system

The VM2000 administrator can use /SHOW-VM-RESOURCES and the INFORMATION=*CPU, VM-IDENTIFICATION=. . . operand to obtain information on the status of the specified VMs' virtual CPUs.

The VM administrator can use /SHOW-VM-ATTRIBUTES and the INFORMATION=*CPU operand to obtain information on the status of the virtual CPUs of his/her VM.

Running virtual CPUs on real CPUs

When scheduling takes place, at runtime a decision is taken regarding the assignment of an operable virtual CPU to a free real CPU from the CPU pool to which the VM belongs (see [page 249](#)). The virtual CPU selected is then started up on the real CPU.

- Ⓢ On S servers scheduling is implemented by the VM2000 hypervisor.
- ⓈQ On SQ servers scheduling is implemented by the Xen hypervisor.

6.1.1.3 Reconfiguration of real and virtual CPUs

Messages that provide information on status changes of real or virtual CPUs are sent to all VM2000 administrators as asynchronous messages and, via routing code “9”, to the authorized BS2000 consoles or applications. See also [page 131](#).

Failure of real CPUs

S On S servers, VM2000 responds to the failure of real CPUs. These CPUs are no longer used by VM2000, the operation continues on the remaining real CPUs. If hot spare CPUs are available, they are attached automatically by VM2000 and replace the failed CPU, see [page 220](#).

CPUs in WFM status (S server with “hot spare CPU”, see [page 214](#)) must be made available again by customer support. An IMPL is then required.

If the real CPU was assigned to a virtual CPU at the time the failure occurred, VM2000 notifies the guest system of this CPU failure. The guest system then executes the requisite failure handling.

SQ On SQ servers, the failure of a real CPU generally results in abnormal termination of operations (server crash (panic)).

The Server Unit must be restarted (an automatic reboot can be set). Defective CPUs or CPU groups then remain detached. The static CPU pools (see [page 34](#)) are configured in such a way that the standard CPU pool as far as possible contains the number of CPUs envisaged for this server model.

VM2000 must be initialized again.

Reconfiguration of real CPUs

If an attached real normal CPU or extra CPU is to be deactivated, the VM2000 administrator can detach it by means of `/DETACH-VM-RESOURCES`. VM2000 operation then continues on the remaining real CPUs.

CPU reconfiguration is rejected if, after the real CPU is rejected, there is not at least one real CPU available in the CPU pool. For details please also refer to the section “[Number of attached real and virtual CPUs in the CPU pool](#)” on [page 228](#).

If a detached real CPU is to be made available again, the VM2000 administrator can attach it again by means of `/ATTACH-VM-RESOURCES`.



If the number of real CPUs decreases, the guest systems remain executable, even if their number of virtual CPUs is greater than the number of real CPUs that are ready for operation (overdimensioned VM).

The expected CPU share of the VMs is adapted, see [page 247](#).

In this case VM2000 does **not** initiate automatic reconfiguration of virtual CPUs. This should be done in the BS2000 guest system by the guest system operator, see the section below.

Reconfiguration of virtual CPUs

Virtual CPUs are reconfigured explicitly in the guest system by means of the BS2000 commands `/ATTACH-DEVICE` and `/DETACH-DEVICE`. `/ATTACH-DEVICE` attaches virtual CPUs, and `/DETACH-DEVICE` detaches them.

Reconfiguration in the guest system makes sense in the following cases, for example:

- In the event of fluctuating loads in the guest system so as only to activate the required CPUs, see also the “Performance Handbook” [17]
- After reconfiguration of real CPUs by VM2000

Example

There is a biprocessor VM, but there is only one real CPU available in the CPU pool (e.g. after reconfiguration of real CPUs by VM2000). In this case, the virtual CPUs run sequentially on the available real CPU. The resulting loss in performance can either be accepted or avoided by means of reconfiguration in the guest system, i.e. in all BS2000 guest systems the multiprocessor level must be adapted by means of `/DETACH-DEVICE UNIT=*CPU(...)` by the same number of CPUs to suit real conditions.

Virtual CPUs are reconfigured implicitly by VM2000 in the following cases:

- When the CPU pool of an active VM is switched with automatic attachment or detachment of virtual CPUs
(`/ASSIGN-VM-TO-CPU-POOL ... , ATTACHED-VM-CPUS=*ADJUST-NUMBER`)
- When the CPU pool is extended or reduced with automatic attachment or detachment of virtual CPUs
(`/SWITCH-VM-CPU ... , TARGET/SOURCE-CPU-POOL=*ADJUST-NUMBER`)

When virtual CPUs in the guest system are reconfigured, the effective CPU share of a VM changes only if the constraints for the CPU share are affected, see [page 243](#).

The reconfiguration of virtual CPUs in the guest system does not have any effect on real CPUs or on virtual CPUs of other guest systems. This also applies to reconfiguration in the monitor system.

The VM2000 commands `/SHOW-VM-ATTRIBUTES` and `/SHOW-VM-RESOURCES` and the BS2000 command `/SHOW-DEVICE-CONFIGURATION (CLASS=*CPU)` indicate the status of the virtual CPUs.

6.1.1.4 High availability with hot spare CPUs

Hot spare CPUs are redundant, fully operable CPUs of an S server that are ready for operation and can be attached automatically **by VM2000** in the event of the failure of a normal or extra CPU and thus take the place of the failed CPU.

Hot spare CPUs thus increase the availability of CPUs as a resource and ensure the continued operation of the server even in the event of the failure of a real normal or extra CPU.

SQ Hot spare CPUs are not available on SQ servers. Cold spare CPUs which (after a CPU has failed) are placed in service when a server has been restarted can be configured on these servers. The former CPU capacity is then available after the restart.

Selected S servers have one (standard) or several hot spare CPUs.

Spare CPUs are not assigned to a CPU pool. These replace a failed CPU in any CPU pool.

Hot spare CPUs in normal operation (without CPU failure)

The real hot spare CPUs of a server are made ready for operation (as far as the hardware is concerned) at startup of the monitor system, but they remain logically disconnected. The number of spare CPUs that are available is indicated in message VMS5050. The real spare CPUs are displayed in VM2000 operation when `/SHOW-VM-RESOURCES INFORMATION=*CPU/*CONFIGURATION` is specified.

In addition to its virtual normal CPUs, each VM also receives virtual spare CPUs. The number of virtual spare CPUs is equal to the number of real spare CPUs.

i If the total number of virtual normal CPUs and virtual spare CPUs is greater than 8 (i.e. greater than the maximum multiprocessor level of a VM), the VM's number of virtual spare CPUs is reduced accordingly.

Guest systems on the VM detect virtual spare CPUs at startup. Multiprocessor guest systems leave virtual spare CPUs detached (status OFF), since they have at least one additional normal CPU available for failure detection.

Mono-guest systems connect a spare CPU so that the operating system can detect the failure of its normal CPU and respond to it (SLEEP status). The number and status of virtual spare CPUs is displayed in the VM-specific part when `/SHOW-VM-ATTRIBUTES INFORMATION=*STD/*CPU` and `/SHOW-VM-RESOURCES INFORMATION=*STD/*CPU` are specified.

Use of hot spare CPUs in the event of CPU failure

In the event of the hardware failure of a real normal or extra CPU (malfunction alert, machine check), the VM2000 hypervisor automatically detaches the defective CPU and attaches a hot spare CPU that is ready for operation. The multiprocessor level of the server remains unchanged by this. The spare CPU is automatically added to the CPU pool to which the failed CPU belonged.

The defective CPU (status WFM) is removed from the CPU pool to which it was assigned. It remains detached until the problem is dealt with by a service engineer. It cannot be attached with `/ATTACH-VM-RESOURCES`. Nor does it become available again after the server is restarted (IMPL, IPL). After it is repaired, the service engineer makes the CPU available again by means of SVP. An IMPL is then required.



Other failures of normal or extra CPUs (e.g. when a CPU gets hung) lead to the CPU being detached (status ERR). In this case no spare CPU is attached. The failed CPU can be attached with `/ATTACH-VM-RESOURCES`. It also becomes available again when the server is restarted (IPL).

A guest system is notified of a CPU error by the VM2000 hypervisor if a virtual CPU of the guest system was active on the failed real normal or extra CPU at the time of the failure. The guest system then detaches this virtual CPU and attaches a virtual spare CPU. The multiprocessor level and power consumption of a VM are not changed by this.

Mono-guest systems process the CPU error on the virtual spare CPU attached at startup. Mono-guest systems can thus continue to function even after being affected by a CPU failure..

The principle of the spare CPU also applies to the monitor system. If there are no more spare CPUs available when the only virtual CPU in the monitor system that is ready for operation fails, provided the restart option is set the VM2000 hypervisor initiates a restart of the monitor system.

6.1.1.5 Increasing capacity with extra CPUs

Extra CPUs are redundant, fully operable CPUs of a server that can be attached by the VM2000 administrator for ongoing VM2000 operation in the event of short-term additional CPU demand being placed on the capacity of the server (Capacity on Demand).

Selected servers are equipped with extras CPUs when the customer requests this.



The use of extra CPUs is regulated by an agreement. Your sales representative will be able to give you details.

Extra CPUs temporarily raise the CPU capacity of the server (see [page 242](#)), the CPU capacity of the CPU pool to which they are assigned, and indirectly the performance of the guest systems as well.

(Real) extra CPUs are managed like real normal CPUs in the CPU pools. They are always assigned to precisely one CPU pool.

Extra CPUs in normal operation

The extra CPUs of a server remain logically disconnected when VM2000 operation is initialized and are added to the standard CPU pool. On S servers the number of available extra CPUs is indicated in message `VMS5053` when the monitor system is started up. The status of the extra CPUs is displayed if `/SHOW-VM-RESOURCES` is specified. Guest systems have nothing to do with extra CPUs.

Use of extra CPUs to raise capacity

If additional CPU capacity is required of the server, the VM2000 administrator can attach extra CPUs by means of `/ATTACH-VM-RESOURCES UNIT=*EXTRA-CPU(...)`.



Extra CPUs must be used in accordance with an agreement. The usage intervals are recorded in the monitor system in a separate log file (RESLOG subsystem). The attachment or detachment of extra CPUs is reported via remote service.



In the event of the failure of extra CPUs on S servers, available hot spare CPUs are attached automatically by VM2000.



In the event of the failure of extra CPUs on SQ servers, installed cold spare CPUs only become available when the server is restarted.

The VM2000 administrator can detach extra CPUs again during VM2000 operation by specifying `/DETACH-VM-RESOURCES UNIT=*EXTRA-CPU(...)`.

Extra CPUs are detached at the latest at VM2000 termination. In a new VM2000 session, extra CPUs have to be attached again, if required.

Distribution of the additional CPU capacity to the VMs

Attaching extra CPUs increases the CPU capacity of the server as a whole. In absolute terms, the server capacity (absolute CPU share) at the disposal of the active VMs of the CPU pool to which the extra CPU is added increases in accordance with the settings for the CPU quotas. If the additional CPU capacity is only to be assigned to one VM or to individual VMs, the CPU quotas of all of the VMs have to be adapted. For details, please also refer to the [section “Planning distribution of the CPU capacity to the VMs” on page 238](#).

Simple example of a biprocessor server (standard CPU pool only)

Initial situation	CPU-QUOTA	Effective CPU share of the VM	Absolute CPU share of the VM
VM1 (BI)	60	60%	1.2
VM2 (BI)	30	30 %	0.6
VM3 (MONO)	10	10 %	0.2
Total	100 ¹	100 %	2.0 CPUs

¹ A total of 100 for the CPU quotas makes it is easy to calculate the CPU shares

When an extra CPU is attached, this results in the following shares:

With an extra CPU	CPU-QUOTA	Effective CPU share of the VM	Absolute CPU share of the VM
VM1 (BI)	60	60%	1.8 ¹
VM2 (BI)	30	30 %	0.9
VM3 (MONO)	10	10 %	0.3
Total	100	100 %	3.0 CPUs

¹ Changes in comparison to the initial situation appear bold

VM2 is to receive all of the additional server capacity:

Adaptation of CPU quotas	CPU-QUOTA	Effective CPU share of the VM	Absolute CPU share of the VM
VM1 (BI)	40	40 %	1.2
VM2 (BI)	53 ¹	53 %	1.6
VM3 (MONO)	7	7 %	0.2
Total	100	100 %	3.0 CPUs

¹ The new CPU quotas and relative CPU shares are calculated in this example on the basis of the relationship between the absolute CPU share of the VM (1.6 for VM2) and the number of CPUs (3).

In complex cases, the formulas and constraints described in the section entitled “Calculating the effective CPU share of a VM” on page 243 must be adhered to.



When setting up the VM with `/CREATE-VM`, ensure that the VM can accept the additional CPU capacity intended for it.

You can do this by setting a sufficiently high multiprocessor level for the `PROCESSOR` operand ($1 \leq \text{PROCESSOR} \leq \text{number of available real normal CPUs}$) so that the absolute CPU share of the VM is lower than the number specified for `PROCESSOR` even after the capacity is increased. In the example on page 223, VM1 and VM2 therefore cannot be set up as mono-VMs (because this would result in absolute CPU shares of 1.2 (VM1) and 1.6 (VM2) after the capacity was increased).

When a VM is even expected to accept the largest possible CPU capacity (overdimensioned VM), i.e. including the CPU capacity of the available extra CPUs, the `PROCESSOR=*EXTRA-AND-NORMAL` operand must be specified when the VM is configured with `/CREATE-VM`, see page 457.

The limit on the CPU utilization of a VM or VM group imposed by `MAX-CPU-UTILIZATION < 100` (see page 240) continues to apply to the nominal CPU capacity of the server. This corresponds to the performance of the real normal CPUs. Attaching or detaching extra CPUs does not change the limit.

The setting `MAX-CPU-UTILIZATION = 100` allows a VM or VM group in normal operation as well as after attaching extra CPUs to utilize the capacity of all attached CPUs.

6.1.2 Managing CPU pools

The concept of CPU pools enables you to combine individual CPUs in an organizational unit. CPU pools have their own name. A CPU pool is assigned individual normal and extra CPUs and also VMs and (on S servers) VM groups. When scheduling takes place, only the VMs and VM groups which are assigned to the CPU pool use the CPUs in the CPU pool.

6.1.2.1 Static CPU pools

Static CPU pools are formed before VM2000 operation is initialized.

Standard CPU pool

The standard CPU pool (*STDPOOL) is the only static CPU pool which is available on all server series.

[SQ] Further static CPU pools are available on SQ servers (*POOL0 , *FOREIGN), see [page 34](#). These are displayed in VM2000 information commands (see [page 33](#)), but are not managed by the VM2000 administrator.

Once VM2000 operation has been initialized, all real CPUs which are available in VM2000 operation are assigned to the standard CPU pool.

[S] Spare CPUs are not assigned to a CPU pool on S servers.

6.1.2.2 Dynamic CPU pools

Dynamic CPU pools can (in addition to the static CPU pools) be configured by the VM2000 administrator after VM2000 READY.

Real normal and extra CPUs can be removed from the standard CPU pool and assigned to a dynamically configured CPU pool. A real normal or extra CPU is thus always belongs to precisely one CPU pool. CPU pools can also be “empty”, i.e. no real CPU and no VM or VM group is assigned to them.

Real spare CPUs are excluded from CPU pools by the management. They do not belong to any CPU pool. See also [section “High availability with hot spare CPUs” on page 220](#).

VMs and VM groups which are assigned to a CPU pool

After VM2000 READY, VMs and (on S servers) VM groups can be assigned to a CPU pool by the VM2000 administrator. A VM or the VMs of a VM group are always assigned to precisely one CPU pool. During scheduling the VM's virtual CPUs are run on real CPUs of the CPU pool.

After VM2000 READY, the monitor VM is assigned to the standard CPU pool. It can also be assigned to a different CPU pool.

Commands for administrating CPU pools

The VM2000 administrator manages a CPU pool in the ADMIN dialog using the monitor VM. The function and privileges of the VM2000 administrator are described in the [section “Operating the whole VM2000 system” on page 186](#).

The table below shows the individual operation groups with the associated VM2000 commands for operating a CPU pool.

Operation group	VM2000 command
Creating a CPU pool	CREATE-VM-CPU-POOL ¹
Adding CPUs to a CPU pool	SWITCH-VM-CPU ¹
Assigning a VM or VM group to a (different) CPU pool	CREATE-VM ¹ CREATE-VM-GROUP ^{1 2} ASSIGN-VM-TO-CPU-POOL ¹ ASSIGN-VM-GROUP-TO-CPU-POOL ^{1 2}
Canceling the assignment of a VM or VM group to a CPU pool	DELETE-VM ³ DELETE-VM-GROUP ^{1 2}
Outputting information on a CPU pool	SHOW-VM-CPU-POOL ¹
Deleting a CPU pool	DELETE-VM-CPU-POOL ¹

¹ For use by the VM2000 administrator only

² only on S servers

³ For use by the VM2000 administrator and the VM administrator of the VM

These operation groups are described in more detail in the sections on [page 231](#).

Schematic overview

The figure below illustrates the options for arranging CPU pools and the assigned virtual machines. In detail, it shows the following:

- The extra CPU CPU06 belongs to the standard CPU pool (it could also be added to a different CPU pool)
- The spare CPU CPU07 does not belong to any CPU pool
- A CPU pool can be assigned no (POOL4), one (POOL2) or multiple VMs or VM groups (POOL3)
- VM groups (GROUP1, GROUP2) are always assigned to precisely one (arbitrary) CPU pool
- The number of CPUs in the CPU pool is independent of the number of VMs which are assigned to this CPU pool (POOL2 / POOL3)

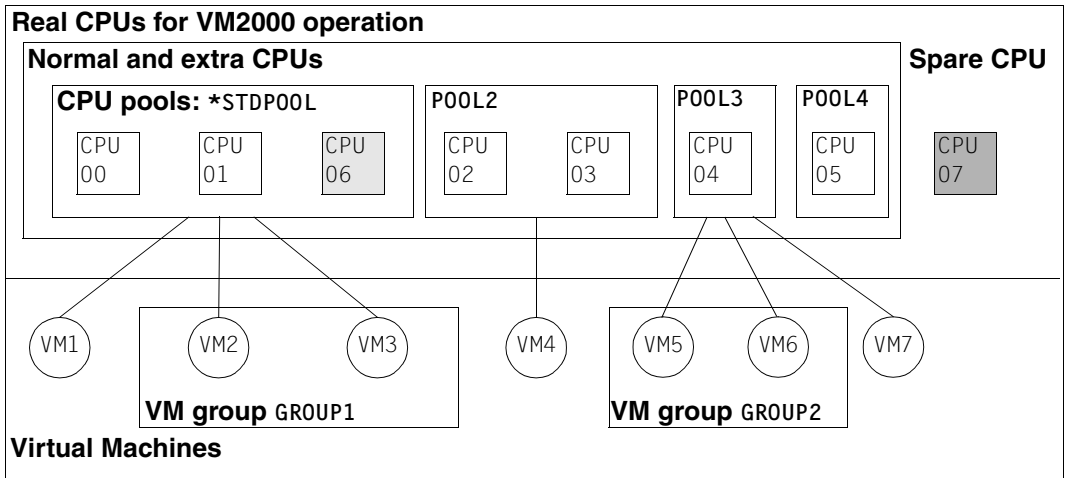


Figure 19: Schematic overview: Possible assignments of VMs to CPU pools

6.1.2.3 Number of attached real and virtual CPUs in the CPU pool

For availability reasons each CPU pool must have at least one attached real CPU if at least one of the following conditions applies:

- The CPU pool is assigned a VM in the RUNNING or IN HOLD (WAIT) status.
- The CPU pool is assigned a VM in the INIT-ONLY, DOWN or MIGRATING status (SQ servers).

The number of attached virtual CPUs of a VM in the RUNNING or IN HOLD (WAIT) status should be **less than or equal to** the number of attached real CPUs of the CPU pool to which the VM is assigned.

This principle guarantees optimum operation of a guest system, i.e. losses in performance (e.g. through scheduling or processor locks) are avoided.

If the principle for a VM in RUNNING or IN HOLD (WAIT) status is no longer observed (overdimensioned VM), the VM will nevertheless continue to run with a multiprocessor level which is too high and the resultant disadvantages.

If the number of attached real CPUs is reduced, in most cases VM2000 guarantees that this principle will be observed. In certain cases (highlighted in **bold print** on the following passages) the VM2000 administrator should cooperate with the guest system operator to ensure the principle is observed.

The following cases can occur:

- Attached real CPUs are deleted from a CPU pool (/SWITCH-VM-CPU):
 - The command is rejected if the last attached real CPU is to be deleted, but a VM in the RUNNING or IN HOLD (WAIT) status is assigned to the CPU pool or, on SQ servers, also in INIT-ONLY, DOWN or MIGRATING status (message/return code VMS4612).
 - Otherwise a check is made for all VMs which are assigned to the CPU pool to see whether the above-mentioned principle can still be observed after the command has been executed. Depending on the ATTACHED-VM-CPUS=*ACTIONS(SOURCE-CPU-POOL=. . .) parameter, the command will be **rejected** (SOURCE-CPU-POOL=*CHECK-NUMBER) or **executed** (SOURCE-CPU-POOL=*ADJUST-NUMBER) if necessary.
 - When SOURCE-CPU-POOL=*ADJUST-NUMBER, VM2000 checks the status of the virtual CPUs of **all VMs** of the source CPU pool and, if required, issues reconfiguration commands for a corresponding number of virtual CPUs in the guest system.

- A VM is started with /START-VM:

If the multiprocessor level m of the VM is greater than the number n of attached real CPUs in the CPU pool, only n virtual CPUs are attached when the guest system is started. The remaining $m - n$ CPUs are identified accordingly by VM2000 (BLOCK status). They can be attached in the guest system later only if further attached real CPUs are added to the CPU pool.

- A VM (RUNNING or IN HOLD (WAIT) status or VM group (with VMs in RUNNING or IN HOLD (WAIT) status) is assigned to a different CPU pool (/ASSIGN-VM-(GROUP-) TO-CPU-POOL):

If the principle is not observed in the new CPU pool, the ATTACHED-VM-CPUS operand determines the further procedure:

- When ATTACHED-VM-CPUS=*CHECK-NUMBER, the command is rejected.
- When ATTACHED-VM-CPUS=*ADJUST-NUMBER, the command is executed if the CPU pool to which the VM or VM group is to be assigned contains at least one attached CPU. The VM or VM group is assigned to the new CPU pool. In addition, VM2000 attempts to detach the excess CPUs (asynchronous, implicit /DETACH-DEVICE in the guest system for a corresponding number of virtual CPUs). These detached CPUs cannot initially be attached again by the guest system; they are identified accordingly by VM2000.
(Status of the virtual CPU in the information commands of VM2000: BLOCK).

- A real CPU is detached (/DETACH-VM-RESOURCES):

VM2000 prevents the last attached real CPU in a CPU pool from being detached if a VM is still active (message/return code VMS4617).

If the principle is no longer observed for a VM in RUNNING or IN HOLD (WAIT) status, the VM nevertheless continues to run with a multiprocessor level which is too high (overdimensioned VM). If necessary, the multiprocessor level should be reduced **in the guest system** by detaching a virtual CPU using /DETACH-DEVICE. A virtual CPU thus detached is then in BLOCK status and can be attached again only when the number of attached real CPUs in the CPU pool has been increased.

- A real CPU fails on an S server and cannot be replaced by a hot spare CPU:

If there are other attached real CPUs in the CPU pool, the procedure to be followed is the same as that described in [“A real CPU is detached \(/DETACH-VM-RESOURCES\):”](#) on page 229.

If the last attached real CPU of a CPU pool fails, VM2000 initiates the following recovery measure to ensure that the VMs and VM groups affected can run:

- In the case of a CPU pool other than the standard CPU pool, the VMs and VM groups affected are assigned to the standard CPU pool
- If the standard CPU pool itself was affected, the attached real CPU with the highest CPU-ID is added to the standard CPU pool. If this CPU is itself the last attached real CPU of a CPU pool, the VMs and VM groups thus affected are also assigned to the standard CPU pool.

The CPU shares of the VMs are recalculated accordingly.

If the number of attached real CPUs in a CPU pool is increased, the principle is always observed. Nevertheless, the VM2000 administrator should take the following into account:

- Attached real CPUs are added to the CPU pool (/SWITCH-VM-CPU)

For all VMs of the target CPU pool whose number of attached virtual CPUs is less than their multiprocessor level, VM2000 cancels the BLOCK status for the relevant number of virtual CPUs in the VM. The virtual CPU can then be attached again in the guest system using /ATTACH-DEVICE.

The ATTACHED-VM-CPUS operand determines the implicit actions of VM2000 for the virtual CPUs of the VMs which are assigned to the target CPU pool:

- When ATTACHED-VM-CPUS=*ACTIONS(TARGET-CPU-POOL=*NONE), no further implicit actions are performed. The number of attached virtual CPUs in the VMs remains unchanged.
 - When ATTACHED-VM-CPUS=*ACTIONS(TARGET-CPU-POOL=*ADJUST-NUMBER), VM2000 checks the status of the virtual CPUs of **all** VMs of the source CPU pool and, if required, issues reconfiguration commands for a corresponding number of virtual CPUs in the guest system.
- Real CPUs are attached (/ATTACH-VM-RESOURCES):

For all VMs of the affected CPU pool whose number of attached virtual CPUs is less than their multiprocessor level, VM2000 cancels the BLOCK status for the relevant number of virtual CPUs in the VM. A detached virtual CPU can then be attached again **in the guest system** using /ATTACH-DEVICE.

- A VM (RUNNING or IN HOLD (WAIT) status) or VM group (with VMs in RUNNING or IN HOLD (WAIT) status) is assigned to a CPU pool which has more attached real CPUs than the previous CPU pool (`/ASSIGN-VM-(GROUP-)TO-CPU-POOL`):

The command is executed. If the multiprocessor level of a VM is greater than the number of attached virtual CPUs, the BLOCK status is canceled for a corresponding number of CPUs. The `ATTACHED-VM-CPUS` operand determines the further procedure:

- `ATTACHED-VM-CPUS=*CHECK-NUMBER` results in no further actions.
- When `ATTACHED-VM-CPUS=*ADJUST-NUMBER`, VM2000 attempts to attach the corresponding number of virtual CPUs (asynchronous, implicit `/ATTACH-DEVICE` in the guest system for a corresponding number of virtual CPUs).

6.1.2.4 Creating a CPU pool

Before a CPU pool can be worked with, it must be created in VM2000. The standard CPU pool is created by VM2000 when the VM2000 is started up. Further CPU pools are created by the VM2000 administrator using `/CREATE-VM-CPU-POOL`. When a CPU pool is created it is given a name.

The maximum number of CPU pools may not exceed the number of the server's real CPUs or the maximum possible number of virtual machines. The number of VMs is hardware-dependent, see [page 457](#).

The maximum number of VMs and the maximum number of CPU pools are also displayed using `/SHOW-VM-RESOURCES INFORMATION=*CONFIGURATION`.

Name of the CPU pool

The name of a CPU pool can be freely selected. It must identify the CPU pool unambiguously. `/CREATE-VM-CPU-POOL` is rejected if a specified CPU pool name has already been assigned.

Recommendations for defining and using the names of CPU pools

The name of a CPU pool should be unique within a VM2000 installation (data center organization) and indicate the purpose of the CPU pool.

The name ranges for VMs, VM groups and CPU pools should be disjunctive.

- SQ On SQ servers the characters #, \$ and @ should not be used in the CPU pool name when a CPU pool is created.

6.1.2.5 Adding CPUs to a CPU pool

After VM2000 READY, all real CPUs which are available for VM2000 operation belong to the standard CPU pool (*STDPPOOL).

The VM2000 administrator can use /SWITCH-VM-CPU to delete real normal and extra CPUs from a CPU pool and add these to another CPU pool.

6.1.2.6 Assigning a VM or VM group to a (different) CPU pool

After a CPU pool has been created, VM groups (S-servers) or VMs which are not members of any VM group can be assigned to this CPU pool.

The VM2000 administrator assigns VMs and VM groups to a CPU pool when the VM is initialized using /CREATE-VM or the VM group is initialized using /CREATE-VM-GROUP (with the CPU-POOL-NAME operand).

After the VM or VM group has been created the VM2000 administrator can assign it to another CPU pool during ongoing operation using /ASSIGN-VM-(GROUP-)TO-CPU-POOL. The VM's or VM group's CPU quota must be reset in the process.



Note on scheduling

The VMs and VM groups which are assigned to a CPU pool are started only on the real CPUs from this CPU pool. The VM's or VM group's CPU quota determines the CPU share of the CPU capacity in the CPU pool, see [page 238](#).

6.1.2.7 Canceling the assignment of a VM or VM group to a CPU pool

The assignment of a VM or a VM group to a CPU pool is also implicitly canceled by the VM administrator when a VM (/DELETE-VM) or VM group (/DELETE-VM-GROUP) is removed.

6.1.2.8 Outputting information on a CPU pool

The VM2000 administrator obtains information on a CPU pool using /SHOW-VM-CPU-POOL. He/She obtains the following information with this command:

- Name of the CPU pool as defined with /CREATE-VM-CPU-POOL
- The scheduling procedure
- The CPUs in the CPU pool and their status
- The VMs assigned to the CPU pool

The VM2000 administrator obtains information on the CPUs in the CPU pool and the VMs which are assigned to a CPU pool using /SHOW-VM-RESOURCES.

6.1.2.9 Deleting a CPU pool

The VM2000 administrator deletes a CPU pool using `/DELETE-VM-CPU-POOL`. A prerequisite here is that no CPU and no VM or VM group are contained in the CPU pool.

The standard CPU pool cannot be deleted.

6.1.2.10 Example of managing CPU pools

The example below illustrates how the VM2000 administrator manages CPU pools on an S server.

```

/show-vm-resources information=*cpu _____ (1)
VM-ID      CPU-POOL  GR-CPU-Q  CPU-Q/  EFF-Q  MAX-U/  EFF-U  PROC
 1 MONITOR  *STDPOOL          30.00/ 42.85 100.00/ 50.00 2(+1)
 2 C1SYS1   *STDPOOL          40.00/ 57.14 100.00/100.00 4(+1)

VM-ID      SCHED  ACT-I  VIRT  CPUS
 1 MONITOR  TS     NO    00(RUN),01(RUN),02(OFF)
 2 C1SYS1   TS     NO    00(RUN),01(RUN),02(RUN),03(RUN),04(OFF)

CPU-ATTR   CPU-POOL      REAL  CPUS
NORMAL     *STDPOOL      00(ATT),01(ATT),02(ATT),03(ATT)
EXTRA      *STDPOOL      04(DET)
SPARE      *STDPOOL      05(DET)

/create-vm-cpu-pool cpu-pool-name=pcust1 _____ (2)
VMS4601 CPU POOL 'PCUST1' CREATED

```

- (1) The command shows the initial situation for this example:
The data center has four real normal CPUs on this server. The existing spare and extra CPUs are unimportant here. A customer VM `C1SYS1` with the attributes shown is running. Time slicing (TS) is used for scheduling, see [page 249](#). The VM2000 administrator also operates the customer VM. The meaning of the output columns is described in more detail in the [section “Planning distribution of the CPU capacity to the VMs” on page 238](#).
- (2) The customer now rents two real normal CPUs for his/her exclusive use. To permit this the VM2000 administrator creates the CPU pool `PCUST1`.

```

/switch-vm-cpu cpu-id=(02,03),from-cpu-pool=*stdpool,to-cpu-pool=pcust1,
  attached-vm-cpus=*actions(source-cpu-pool=*adjust-number) _____ (3)
VMS4608 CPU '02' SWITCHED FROM CPU-POOL '*STDPool' TO CPU-POOL 'PCUST1'
VMS4608 CPU '03' SWITCHED FROM CPU-POOL '*STDPool' TO CPU-POOL 'PCUST1'
VMS4621 IMPLICIT DETACH-DEVICE FOR VIRTUAL CPUS ON
      VIRTUAL MACHINE (2,C1SYS1) STARTED _____ (4)
% VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (2,C1SYS1),
      CODE 80FFFFFF, CPU 03
% VMS2011 CPU 03 OF VIRTUAL MACHINE (2,C1SYS1) STOPPED
% VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (2,C1SYS1),
      CODE 80FFFFFF, CPU 02
% VMS2011 CPU 02 OF VIRTUAL MACHINE (2,C1SYS1) STOPPED
/show-vm-cpu-pool cpu-pool-name=*all,information=*cpu _____ (5)
CPU-POOL SCHED CPU-ATTR REAL CPUS
*STDPool   TS   NORMAL   00(ATT),01(ATT)
              EXTRA   04(DET)
PCUST1     -   NORMAL   02(ATT),03(ATT)
              EXTRA   NONE
/assign-vm-to-cpu-pool vm-id=c1sys1,to-cpu-pool=pcust1,cpu-quota=80 _____ (6)
VMS4618 CPU POOL ASSIGNMENT OF VIRTUAL MACHINE (2,C1SYS1) CHANGED
      FROM CPU POOL '*STDPool' TO CPU POOL 'PCUST1'

```

- (3) The real CPUs 02 and 03 are switched from the standard CPU pool to the customer CPU pool PCUST1.
- (4) The SOURCE-CPU-POOL=*ADJUST-NUMBER operand causes VM2000 to check, for all VMs in the source CPU pool, whether the number of virtual CPUs which they possess remains greater than or equal to the remaining number of real CPUs. In this case VM2000 initiates a detach operation for the virtual CPUs 02 and 03 in the customer system as only two real CPUs are left in the standard CPU pool but four virtual CPUs are running in the customer VM C1SYS1. Asynchronous messages are marked with % here.
- (5) The information output for the CPU pools shows the distribution of real normal and extra CPUs.
- (6) The customer VM C1SYS1 is assigned to the customer CPU pool PCUST1. The VM's CPU quota is reset in the process.

Note

With this command the default value ATTACHED-VM-CPUS=*CHECK-NUMBER is used to check whether the number of attached virtual CPUs of the VM (2) is less than or equal to the number of attached real CPUs of the CPU pool (2). If this were not the case, the command would be rejected.

```

/ create-vm memory-size=128,vm-name=c1sys2,processor=*bi,cpu-quota=20,
      cpu-pool-name=pcust1 _____ (7)
VMS3050 VIRTUAL MACHINE (3,C1SYS2) CREATED
VMS4615 VIRTUAL MACHINE (3,C1SYS2) ASSIGNED TO CPU POOL 'PCUST1'
/ show-vm-cpu-pool cpu-pool-name=pcust1,information=*vm _____ (8)
CPU-POOL SCHED VM-ID          VM-GROUP ATT PROC
PCUST1    DC    2 C1SYS1          2    4(+1)
           3 C1SYS2          0    2(+1)
/ add-vm-devices vm-id=c1sys2,
      units=*pubset-devices(pubset=*by-pubres-device(unit=e05c)) _____ (9)
VMS4005 DEVICE 'E05C' ASSIGNED TO VIRTUAL MACHINE (3,C1SYS2)
VMS4005 DEVICE 'E05D' ASSIGNED TO VIRTUAL MACHINE (3,C1SYS2)
VMS4005 DEVICE 'E05E' ASSIGNED TO VIRTUAL MACHINE (3,C1SYS2)
VMS4220 PUBSET '4IVX' WITH PUBRES DEVICE 'E05C' ASSIGNED TO VM (3,C1SYS2)
/ add-vm-devices units=(cj,ck),vm-id=c1sys2
VMS4005 DEVICE 'CJ' ASSIGNED TO VIRTUAL MACHINE (3,C1SYS2)
VMS4005 DEVICE 'CK' ASSIGNED TO VIRTUAL MACHINE (3,C1SYS2)
/ start-vm ipl-unit=e05c,vm-id=c1sys2
% VMS2023 CPU 01 OF VIRTUAL MACHINE (3,C1SYS2) STARTED
% VMS2023 CPU 02 OF VIRTUAL MACHINE (3,C1SYS2) STARTED
% VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (3,C1SYS2),
      CODE 80FFFFFF, CPU 02
% VMS2011 CPU 02 OF VIRTUAL MACHINE (3,C1SYS2) STOPPED
% VMS2050 GUEST SYSTEM ON VM (3,C1SYS2) READY
/ show-vm-cpu-pool cpu-pool-name=pcust1,information=*summary _____ (10)
CPU-POOL SCHED REAL-NORMAL-CPUS REAL-EXTRA-CPUS VIRT-CPUS
           ALL ATT DET OFF  ALL ATT DET OFF  RUN
PCUST1    TS    2  2  0  0    0  0  0  0    4

```

- (7) The customer wants to have a further VM so that the two customer VMs can share the capacity of the rented CPUs in a ratio of 4:1. For this purpose the VM2000 administrator creates the customer VM C1SYS2 with a CPU quota of 20 and assigns it to the customer CPU pool PCUST1.
- (8) The information output for the customer CPU pool PCUST1 shows that initially only the customer VM C1SYS1 is running. As for two attached virtual CPUs the VM also has two attached real CPUs available in the CPU pool, a fixed assignment is implemented during scheduling (dedicated CPUs, DC), see [page 249](#).
- (9) The customer system is started. Here the virtual spare CPU 02 is also briefly started and then stopped again.
- (10) The information output for the customer CPU pool PCUST1 shows a concise overview of the real and virtual CPUs. As four virtual CPUs are now running on two real CPUs of the CPU pool, time slicing is automatically used for scheduling (TS).

```
/show-vm-resources information=*cpu,  
vm-id=*by-cpu-pool(cpu-pool-name=pcust1) _____ (11)
```

VM-ID	CPU-POOL	GR-CPU-Q	CPU-Q/	EFF-Q	MAX-U/	EFF-U	PROC
2	C1SYS1	PCUST1	80.00/	40.00	100.00/	50.00	4(+1)
3	C1SYS2	PCUST1	20.00/	10.00	100.00/	50.00	2(+1)

VM-ID	SCHED	ACT-I	VIRT	CPUS
2	C1SYS1	TS	NO	00(RUN),01(RUN),02(BLOCK),03(BLOCK),04(OFF)
3	C1SYS2	TS	NO	00(RUN),01(RUN),02(OFF)

```
/shutdown-vm vm-id=c1sys2,action=*shutdown(quiet-time=0) _____ (12)
```

```
VMS2054 SHUTDOWN OF GUEST SYSTEM ON VM (3,C1SYS2) INITIATED
% VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (3,C1SYS2),
    CODE 80FFFFFF, CPU 01
% VMS2033 'EXC0557 SHUTDOWN PROCESSING COMPLETED' FROM VM (3,C1SYS2) VIA SVP
% VMS2051 GUEST SYSTEM ON VM (3,C1SYS2) DOWN; REASON: SHUTDOWN
% VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (3,C1SYS2),
    CODE 80FFFFFF, CPU 00
```

```
/show-vm-cpu-pool cpu-pool-name=pcust1,information=*all
```

CPU-POOL	SCHED	CPU-ATTR	REAL	CPUS	_____
PCUST1	DC	NORMAL	02(ATT),03(ATT)		(13)
		EXTRA	NONE		

CPU-POOL	SCHED	VM-ID	VM-GROUP	ATT	PROC
PCUST1	DC	2 C1SYS1		2	4(+1)
		3 C1SYS2		0	2(+1)

```
/delete-vm vm-id=c1sys2 _____ (14)
```

```
VMS4009 DEVICE 'CJ' REMOVED FROM VIRTUAL MACHINE (3,C1SYS2)
VMS4009 DEVICE 'CK' REMOVED FROM VIRTUAL MACHINE (3,C1SYS2)
VMS4009 DEVICE 'E05C' REMOVED FROM VIRTUAL MACHINE (3,C1SYS2)
VMS4009 DEVICE 'E05D' REMOVED FROM VIRTUAL MACHINE (3,C1SYS2)
VMS4009 DEVICE 'E05E' REMOVED FROM VIRTUAL MACHINE (3,C1SYS2)
VMS4619 VIRTUAL MACHINE (3,C1SYS2) REMOVED FROM CPU POOL 'PCUST1'
VMS4024 VIRTUAL MACHINE (3,C1SYS2) TERMINATED
```

- (11) The information output for the VMs of the customer CPU pool PCUST1 shows that the two VMs are sharing the available CPU capacity in the required ratio (planned values).
- (12) After a certain time, the VM2000 administrator shuts down the guest system on VM C1SYS2.
- (13) As a result the procedure for fixed CPU assignment is automatically used again for scheduling (dedicated CPUs, DC), see [page 249](#).
- (14) The VM is terminated and removed from the customer CPU pool PCUST1.

```

/assign-vm-to-cpu-pool vm-id=c1sys1,to-cpu-pool=*stdpool,cpu-quota=40 - (15)
VMS4618 CPU POOL ASSIGNMENT OF VIRTUAL MACHINE (2,C1SYS1) CHANGED
      FROM CPU POOL 'PCUST1' TO CPU POOL '*STDPOOL'
/switch-vm-cpu cpu-id=*all,from-cpu-pool=pcust1,to-cpu-pool=*stdpool,
  attached-vm-cpus=*actions(target-cpu-pool=*adjust-number) ----- (16)
VMS4608 CPU '02' SWITCHED FROM CPU-POOL 'PCUST1' TO CPU-POOL '*STDPOOL'
VMS4608 CPU '03' SWITCHED FROM CPU-POOL 'PCUST1' TO CPU-POOL '*STDPOOL'
VMS4620 IMPLICIT ATTACH-DEVICE FOR VIRTUAL CPUS ON
      VIRTUAL MACHINE (2,C1SYS1) STARTED ----- (17)
% VMS2023 CPU 02 OF VIRTUAL MACHINE (2,C1SYS1) STARTED
% VMS2023 CPU 03 OF VIRTUAL MACHINE (2,C1SYS1) STARTED
/show-vm-cpu-pool cpu-pool-name=pcust1,information=*all ----- (18)
CPU-POOL SCHED CPU-ATTR REAL CPUS
PCUST1      -    NORMAL    NONE
              EXTRA    NONE

CPU-POOL SCHED VM-ID          VM-GROUP ATT PROC
PCUST1      -          NONE

/delete-vm-cpu-pool cpu-pool-name=pcust1 ----- (19)
VMS4605 CPU POOL 'PCUST1' DELETED

```

- (15) The VM `C1SYS1` is (once more) assigned to the standard CPU pool. The CPU quota is reset to the earlier value in the process.
- (16) All real CPUs of the customer CPU pool `PCUST1` are once more switched to the standard CPU pool.
- (17) The `TARGET-CPU-POOL=*ADJUST-NUMBER` operand causes VM2000 to check, for all VMs in the target CPU pool, whether the number of virtual CPUs which they possess is less than or equal to the new number of real CPUs. In this case VM2000 initiates an attach operation for virtual CPUs `02` and `03` in the customer system as four real CPUs will be in the standard CPU pool but only two of the four virtual CPUs created are running in the customer VM `C1SYS1`.
- (18) Neither CPUs nor VMs are contained in the CPU pool `PCUST1`. It can be deleted.
- (19) The CPU pool `PCUST1` is deleted.
The initial situation for the example has now been reestablished.

6.1.3 Planning distribution of the CPU capacity to the VMs

The VM2000 administrator can plan how to distribute the CPU capacity to the VMs and VM groups in detail and modify this during ongoing operation.

Parameters for distributing the CPU capacity

Under VM2000 there are two parameters which the VM2000 administrator can use to plan and influence the long-term distribution of the available CPU capacity to the VMs:

- privileges of the VM

The CPU quota is a specification made by the VM2000 administrator for the CPU share of the **CPU capacity of the CPU pool** which a VM group or a VM that does not belong to a VM group has in comparison to the other VMs which are assigned to the same CPU pool directly or as members of a VM group.

For VMs which do not belong to a VM group, the CPU quota is specified in the CPU-QUOTA operand of /CREATE-VM or /MODIFY-VM-ATTRIBUTES.

For VM groups (S servers), the CPU quota is specified in the CPU-QUOTA operand of /CREATE-VM-GROUP or /MODIFY-VM-GROUP-ATTRIBUTES.

For VMs which are members of a VM group, the member CPU quota determines the share the VM has of the VM group's CPU share. The member CPU quota is specified in the MEMBER-CPU-QUOTA operand of /CREATE-VM or /MODIFY-VM-ATTRIBUTES.

To permit a better comparison, the CPU quotas set are used (on the basis of idealized conditions) to calculate a standardized CPU share for the VM of the **server's CPU capacity**, which is referred to as the **effective CPU share of the VM (EFF-Q)**, see [“Calculating the effective CPU share of a VM” on page 243](#).



For the definition of the CPU capacity of a server, see [page 242](#).

The effective CPU share of a VM is displayed in the CPU-Q / EFF-Q column of the VM2000 information commands in addition to the CPU quota set by means of a command.

- The maximum CPU utilization

The maximum CPU utilization restricts the CPU share of a VM or of a VM group. It is specified as the percentage of the **available CPU capacity in VM2000 operation** (number of available real normal CPUs) which is granted for the VM group or VM. The maximum CPU utilization thus restricts the CPU utilization of a VM or of a VM group which would otherwise take up too much CPU time.

For VMs the maximum CPU utilization is specified in the `MAX-CPU-UTILIZATION` operand of `/CREATE-VM` or `/MODIFY-VM-ATTRIBUTES`.

For VM groups the maximum CPU utilization is specified in the `MAX-CPU-UTILIZATION` operand of `/CREATE-VM-GROUP` or `/MODIFY-VM-GROUP-ATTRIBUTES`.

A standardized value, known as the **effective maximum CPU utilization of the VM (EFF-U)**, is calculated (on the basis of idealized conditions) from the maximum CPU utilization set, see [“Calculating the effective CPU share of a VM” on page 243](#).

The effective maximum CPU utilization of a VM is displayed in the `CPU-U / EFF-U` column of the VM2000 information commands in addition to the maximum CPU utilization set by means of a command.

Constraints for planning

Idealized conditions are assumed for calculating the effective CPU share and the effective maximum CPU utilization for the VM:

- All real normal CPUs are attached (extra and spare CPUs are ignored)
- All virtual machines are running at peak load on all virtual CPUs, i.e. they are making full use of their CPU time slices
- Performance losses through scheduling are ignored

In ongoing VM2000 operation, `/SHOW-VM-ATTRIBUTES/-RESOURCES` display the planned values under the **idealized** conditions.

In ongoing VM2000 operation, `/SHOW-VM-STATUS` displays the planned values under the **current** conditions and the associated results measurement results, see [page 248](#).

Specifying the parameters for distributing the CPU capacity

The VM2000 administrator has the following options for specifying the two parameters, CPU quota and maximum CPU utilization, which determine the CPU shares of a VM:

- Planning the CPU share of the VMs via the CPU quota in the case of unrestricted CPU utilization of the VMs and VM groups (`MAX-CPU-UTILIZATION = 100`). This setting is designed for optimum server load. In the case of peak operation in all VMs, the effective CPU share of a VM is determined using the formulae in the section [“Calculating the effective CPU share of a VM” on page 243](#). During off-peak operation, a VM with a low effective CPU quota can still take up a large amount of CPU capacity.

Example (without VM groups)

VM (mono server)	CPU-QUOTA	MAX-CPU-UTILIZATION	Effective CPU share of the VM ¹
1	10	100	20 %
2	15	100	30 %
3	25	100	50 %

¹ At peak load the CPU would always be assigned here

- Planning the CPU share of **all** VMs via the maximum CPU utilization of the VMs and VM groups (`MAX-CPU-UTILIZATION < 100`). If the maximum CPU utilization of the VMs is set such that its sum is less than or equal to 100 (minus the utilization by the VM2000 hypervisor on S servers), then the maximum CPU utilization of a VM in peak operation corresponds to the maximum CPU utilization of a VM in peak operation also corresponds to its CPU share (in percent). The load on the server depends on the sum of the maximum CPU utilization of the VMs and the load profiles of the VMs. In this case, the CPU quota is relevant for the prioritization, but not for the possible CPU utilization.

Example (without VM groups)

VM (mono server)	CPU-QUOTA	MAX-CPU-UTILIZATION	Effective CPU share of the VM ¹
1	10	20	20 %
2	15	25	25 %
3	25	40	40 %

¹ At peak load the CPU would not always be assigned here

- Planning the CPU share of **selected** VMs via the maximum CPU utilization ($MAX-CPU-UTILIZATION \leq 100$).

In this case, the maximum CPU utilization restricts the VMs which unintentionally take up a lot of CPU capacity. Nevertheless, the server remains optimally utilized during peak operation. For a restricted VM to be able to utilize its planned CPU capacity, the CPU quotas of the VMs must be set such that the expected CPU share of the VM corresponds to its maximum CPU utilization.

Example (without VM groups)

VM (mono server)	CPU-QUOTA	MAX-CPU-UTILIZATION	Effective CPU share of the VM ¹
1	10	100	30 %
2	15	100	45 %
3	25	25	25 %

¹ At peak load the CPU would always be assigned here

Example (with one VM group, S server)

VM (mono server)	CPU-QUOTA	MAX-CPU-UTILIZATION	Effective CPU share of the VM ¹
1	10	100	30 %
2	15	100	45 %
VM group GROUP1	25 MEMBER-CPU-QUOTA	25	(25 %) ²
3	80	25	20 %
4	20	25	5 %

¹ At peak load the CPU would always be assigned here

² Distributed over the two VMs of the VM group

Recommendation

The CPU shares of the VMs and VM groups should continue to be regulated via the CPU quota in order to achieve optimum utilization during peak operation.

In the event of VMs with unacceptably high CPU utilization (or if a performance limit has been defined for the CPU time), this can be restricted by the maximum CPU utilization. Measurements have shown that the ratio of “load on the guest system / MAX-CPU-UTILIZATION” should be under 0.75, otherwise the response time behavior of the guest system concerned deteriorates considerably.

Data centers which provide a certain CPU capacity for a VM (expressed in RPF) must note that the nominal capacity of the server is reduced through the use of VM2000, depending on the number and multiprocessor level of the VMs. MAX-CPU-UTILIZATION=20 is thus less than 20% of the server capacity!

i The multiprocessor level of VMs with greatly restricted performance (effective CPU share < 1, e.g. because of `MAX-CPU-UTILIZATION < 1`) which is selected should also be low (`MONO` or at most `BI` (on servers without spare CPU)) to ensure that the guest system can operate without a hitch.

CPU capacity and CPU share of a VM

In addition to the CPU quota and maximum CPU utilization parameters, the CPU share of a VM also depends on the CPU capacity of the server or CPU pool to which the VM is assigned.

The **CPU capacity of a server** is the CPU capacity of the real (normal) CPUs which are available in VM2000 operation (see [page 213](#)), irrespective of whether they are attached or detached. It is expressed in terms of the number of CPUs.

i Attached extra CPUs increase the server's CPU capacity temporarily (see [page 222](#)). They are not taken into account for the calculations in this section.

The **CPU capacity of a CPU pool** is defined as the share of the real normal CPUs of the CPU pool (regardless of whether they are attached or detached) in the CPU capacity of the server according to the formula

$$\text{CPU capacity of the CPU pool} = \frac{\text{Number of real normal CPUs of the CPU pool} \times 100}{\text{Number of available real normal CPUs}} \%$$

Example

A CPU pool with two real normal CPUs on a quadro-server has a CPU capacity which is 50% of that of the server.

A virtual machine receives a (theoretical) share of the CPU capacity of the CPU pool to which the VM is assigned.

For planning purposes, the CPU capacity of the CPU pool (and the parameters set, see [page 238](#)) are used to calculate the CPU share of a VM in the server's CPU capacity, see ["Calculating the effective CPU share of a VM" on page 243](#).

Calculating the effective CPU share of a VM

Theoretical values for the distribution of the available CPU capacity to the virtual machines are calculated under the idealized conditions mentioned on [page 239](#) from the values for the CPU quota and the maximum CPU utilization set by the VM2000 administrator.

Case 1: Not taking into account VM groups

The **effective CPU share of the VM (EFF-Q)** is calculated using the formula:

$$(F1) \quad \text{EFF-Q} = \frac{\text{CPU-QUOTA of the VM} \times \text{CPU capacity of the CPU pool}}{\text{Sum of the CPU-QUOTAs of all VMs of the CPU pool}} \%$$

Example for a CPU pool with 2 normal CPUs bei 4 verfügbaren Normal-CPU's

	VM1 (BI)	VM2 (BI)	VM3 (MONO)
CPU-QUOTA (sum: 150)	90.00	50.00	10.00
Effective CPU share of the VM (EFF-Q) ¹ (CPU capacity of CPU pool: 50 %)	30.00 %	16.67 %	3.33 %

¹ Assuming that the maximum utilization of VMs is not restricted

The effective CPU share of a VM may be restricted by the following constraints:

$$(N1) \quad \text{EFF-Q} \leq \text{Maximum CPU utilization of the VM (by means of command)} \%$$

$$(N2) \quad \text{EFF-Q} \leq \frac{\text{Number of real CPUs of the CPU pool} \times 100}{\text{Number of available real normal CPUs}} \%$$

$$(N3) \quad \text{EFF-Q} \leq \frac{\text{Number of virtual CPUs of the VM (multiprocessor level)} \times 100}{\text{Number of available real normal CPUs}} \%$$

The **effective maximum CPU utilization of the VM (EFF-U)** is the minimum value for these constraints and restricts the effective CPU share of the VM:

$$(F2) \quad \text{EFF-Q} \leq \text{EFF-U} = \text{Minimum}((N1), (N2), (N3))$$

If the effective CPU share of one or more VMs (according to formula (F1)) is restricted by the effective maximum CPU utilization of the VM, the remaining CPU share is distributed to the unrestricted VMs in accordance with their CPU quota:

$$(F3) \quad \text{EFF-Q} = \frac{\text{CPU-QUOTA of the VM} \times (\text{CPU capacity of the CPU pool} - \sum \text{CV})}{\text{Sum of the CPU-QUOTAs of all unrestricted VMs of the CPU pool}} \%$$

where $\sum \text{CV}$ = Sum of the effective CPU shares of the VMs of the CPU pool whose effective CPU share is restricted by (N1) or (N3).

Example for a CPU pool with 2 normal CPUs and 4 available normal CPUs with restricted VMs

	VM1 (BI)	VM2 (BI)	VM3 (MONO)
CPU-QUOTA (sum: 150)	10.00	50.00	90.00
EFF-Q acc. to formula (F1) (CPU capacity of CPU pool: 50 %)	3.33 %	16.67 %	30.00 %
MAX-CPU-UTILIZATION	100.00	15.00	60.00
Effective maximum utilization of the VM (EFF-U)	50.00 % because of (N2)	15.00 % because of (N1)	25.00 % because of (N3)
Effective CPU share of the VM restricted by EFF-U	10.00 % acc. to formula (F3)	15.00 % acc. to formula (F2)	25.00 % acc. to formula (F2)

The CPU share is calculated in the VM2000 hypervisor. In the example, VM3 was given the maximum possible CPU share; the rest of the CPU capacity of the CPU pool remains for VM1 and VM2.

(In practice, the above example would mean that VM3 should be assigned the CPU capacity of one of the two CPUs available in the CPU pool (50% of the pool's CPU capacity = 25% of the server's CPU capacity = one CPU), while VM1 and VM2 with their two virtual CPUs each would share the CPU capacity of the other CPU in the CPU pool.)

Case 2: Taking into account VM groups (S servers)

The **effective CPU share of the VM (EFF-Q)** is calculated using the formula:

(F1a)	$\text{EFF-Q (VM or VM group)} = \frac{\text{CPU-QUOTA (VM w/o VM group)} \times \text{CPU capacity of the CPU pool}}{\text{Sum of CPU-QUOTA of all VMs w/o group and VM gr. of CPU pool}} \%$
(F1b)	$\text{EFF-Q (VM in the group)} = \frac{\text{MEMBER-CPU-QUOTA of the VM} \times \text{EFF-Q of the VM group}}{\text{Sum of the MEMBER-CPU-QUOTAs of all VMs in the group}} \%$

Example for a CPU pool with 2 normal CPUs on a quadro-server

VMs	VM1 (BI)	VM group GROUP1	
		VM2 (BI)	VM3 (MONO)
CPU-QUOTA (sum: 150)	90.00	60.00 (effective CPU share of the VM group: 60*50/150 = 20 %)	
MEMBER-CPU-QUOTA (in VM group)	–	50.00	10.00
Effective CPU share of the VM (EFF-Q) ¹ acc. to formulae (F1a) and (F1b) (CPU capacity of CPU pool: 50 %)	30.00 %	16.67 % ²	3.33 % ²

¹ Assuming that the maximum utilization of VMs is not restricted

² Ignoring “load balancing within the VM group”, see [page 252](#)

The effective CPU share of a VM may be restricted by the following constraints:

(N1a)	$\text{EFF-Q} \leq \text{Maximum CPU utilization of the VM (by means of command)} \%$
(N1b)	$\text{EFF-Q} \leq \text{Maximum CPU utilization of the VM group (by means of command)} \%$
(N2)	$\text{EFF-Q} \leq \frac{\text{Number of real CPUs of the CPU pool} \times 100}{\text{Number of real CPUs of the server}} \%$
(N3)	$\text{EFF-Q} \leq \frac{\text{Number of virtual CPUs of the VM (multiprocessor level)} \times 100}{\text{Number of real CPUs of the server}} \%$

The **effective maximum CPU utilization of the VM (EFF-U)** is the minimum value for these constraints and restricts the effective CPU share of the VM:

(F2)	$\text{EFF-Q} \leq \text{EFF-U} = \text{Minimum}((\text{N1a}), (\text{N1b}), (\text{N2}), (\text{N3}))$
------	---

If the effective CPU share of one or more VMs (according to formulae (F1) and (F1b)) is restricted by the effective maximum CPU utilization of the VM, the remaining CPU share is distributed over the unrestricted VMs in accordance with their CPU quota:

$$(F3) \quad \text{EFF-Q} = \frac{\text{CPU-QUOTA of the VM} \times (\text{CPU capacity of the CPU pool} - \sum \text{CV})}{\text{Sum of the CPU-QUOTAs of all unrestricted VMs of the CPU pool}} \%$$

where $\sum \text{CV}$ = Sum of the effective CPU shares of the VMs of the CPU pool whose effective CPU share is restricted by (N1) or (N3).

Example for a CPU pool with 2 normal CPUs on a quadro-server with restricted VMs

VMs	VM group GROUP1		
	VM1 (BI)	VM2 (BI)	VM3 (MONO)
CPU-QUOTA (sum: 100)	10.00	90.00 (effective CPU share of the VM group: 90*50/100 = 45 %)	
MEMBER-CPU-QUOTA (in VM group)	–	50.00	90.00
Effective CPU share of the VM (EFF-Q) acc. to formulae (F1a) and (F1b) (CPU capacity of CPU pool: 50 %)	5.00 %	16.07 %	28.93 %
MAX-CPU-UTILIZATION (VM group)	–	50.00	
MAX-CPU-UTILIZATION (VM)	100.00	15.00	60.00
Effective maximum utilization (EFF-U)	50.00 % because of (N2)	15.00 % because of (N1a)	25.00 % because of (N3)
Effective CPU share of the VM with constraints (EFF-Q)	10.00 % acc. to formula (F3)	15.00 % acc. to formula (F2)	25.00 % acc. to formula (F2)

The CPU share is calculated in the VM2000 hypervisor. In the example, VM3 was given the maximum possible CPU share; the rest of the CPU capacity of the CPU pool remains for VM1 and VM2.

(In practice, the above example would mean that VM3 should be assigned the CPU capacity of one of the two CPUs available in the CPU pool (50% of the pool's CPU capacity = 25% of the server's CPU capacity = one CPU), while VM1 and VM2 with their two virtual CPUs each would share the CPU capacity of the other CPU in the CPU pool.)

Changes to the effective CPU share of a VM

The VM2000 administrator can explicitly control the effective CPU share of a VM by changing the CPU quota and the maximum CPU utilization, see [page 238](#). The relevant values are displayed using `/SHOW-VM-RESOURCES INFORMATION=*CPU`, see [page 579](#).

The effective CPU share of the VM changes implicitly during ongoing operation when the following changes occur:

- The CPU quota of another VM or VM group which is assigned to the same CPU pool changes
- The maximum utilization of the VM or VM group changes, which affects the VM's effective CPU share
- The number of normal CPUs which belong to the same CPU pool changes (regardless of their reconfiguration status)
- The VM is assigned to another CPU pool
- The number of VMs which are assigned to the same CPU pool changes
- The number of VMs which belong to the same CPU pool changes
- The CPU quota of the VM group to which the VM belongs changes

The effective CPU share of a VM does not change when **virtual** CPUs are reconfigured in the guest system.

When **real** CPUs are reconfigured, the number of active CPUs in VM2000 changes, but not the CPU capacity of the server under the assumed idealized conditions. The effective CPU share and the effective maximum utilization of the VMs do not change.

The attachment of an extra CPU does not cause the effective CPU share of the VMs to change.

Output of the current planned and measured values for the CPU share of the VMs

The `/SHOW-VM-STATUS` command (see [page 607](#)), in contrast to the idealized conditions assumed for planning, uses the current values of the active objects in VM2000 operation.

The active objects in VM2000 operation are:

- the attached real normal and **extra** CPUs
- the active VMs
- the active virtual CPUs

As with the idealized conditions, it is assumed that all active virtual machines are running at peak load on all active virtual CPUs, i.e. they are making full use of their CPU time slices.

In the VM-specific information block, `/SHOW-VM-STATUS INFORMATION=*STD/*TIMER` displays, in addition to the CPU quota set (`CPU-Q`), the planned values for the CPU share of the VMs (`CUR-Q`) calculated on the basis of the active objects in VM2000 operation and the maximum CPU utilization of the VMs (`CUR-U`), see [page 614](#).

The calculation is based on the formulae `EFF-Q` and `EFF-U`, which are shown on [page 243](#), but related to the active objects in VM2000 operation.

By way of comparison, the command displays the current measured values for the CPU shares of the VMs in absolute terms and as percentage values.

6.1.4 Scheduling procedures

When scheduling a VM, operable virtual CPUs of a VM are started up on free real CPUs from the CPU pool to which the VM is assigned.

Here the CPU pools CPU capacity is distributed to the operable virtual machines in the most efficient manner in accordance with the settings for the VMs (see [page 238](#)).

S On S servers scheduling is implemented by the VM2000 hypervisor in one of two different procedures:

- Fixed CPU assignment (dedicated CPUs)
- CPU assignment in the time-slicing procedure

SQ On SQ servers scheduling is implemented by the Xen hypervisor in a manner similar to time slicing.

The current scheduling procedure for a VM is output using `/SHOW-VM-ATTRIBUTES, /SHOW-VM-RESOURCES, /SHOW-VM-CPU-POOL` with `INFORMATION=*CPU`.

6.1.4.1 Fixed CPU assignment (dedicated CPUs, S servers)

If the number of attached real CPUs in a CPU pool is **greater than or equal to** the sum of the attached virtual CPUs of all active VMs (RUNNING status) which are assigned to this CPU pool, VM2000 permanently assigns precisely one CPU to each virtual CPU of a VM. These CPUs are referred to as dedicated CPUs.

In the VM2000 information commands this scheduling procedure is referred to as DC (dedicated CPUs).

Fixed CPU assignment is ideal if there is a sufficient number of real CPUs since each virtual CPU always runs on one and the same real CPU.

No losses in performance occur as a result of cache reestablishment when the virtual CPU is switched or the context is changed to the VM2000 hypervisor in the idle state (when `VM-ACTIVE-IDLE=*AT-DEDICATED-CPU`, see [page 147](#)).

However, if a real CPU is not used, it remains in the idle state. It is not assigned to another VM which is waiting outside the CPU pool.

For VMs in this scheduling procedure, the `VM-ACTIVE-IDLE` attribute determines whether a VM still retains control over a real CPU if the VM's virtual CPU which is running on it is inactive (interruptible wait state), see [page 147](#). The setting for `VM-ACTIVE-IDLE` is displayed using `/SHOW-VM-ATTRIBUTES/-RESOURCES INFORMATION=*CPU`.

If a limit for CPU utilization is also set for the VM, this attribute is not effective. `/SHOW-VM-STATUS INFORMATION=*SCHEDULE` displays whether this attribute is currently effective, see [page 619](#).

If the number of attached real CPUs drops below the required number (e.g. through reconfiguration or addition of a new VM), VM2000 switches to time slicing for the purpose of CPU distribution.

Example

The virtual machines `SYSTEM1` and `SYSTEM2` are assigned to the CPU pool `POOL1`. Both VMs are running with two virtual CPUs each. As four real CPUs are attached in the CPU pool, the real CPUs are permanently assigned to the virtual CPUs.

Only the effective maximum utilization of the VM is taken into account for scheduling. The CPU quota is of no significance here.

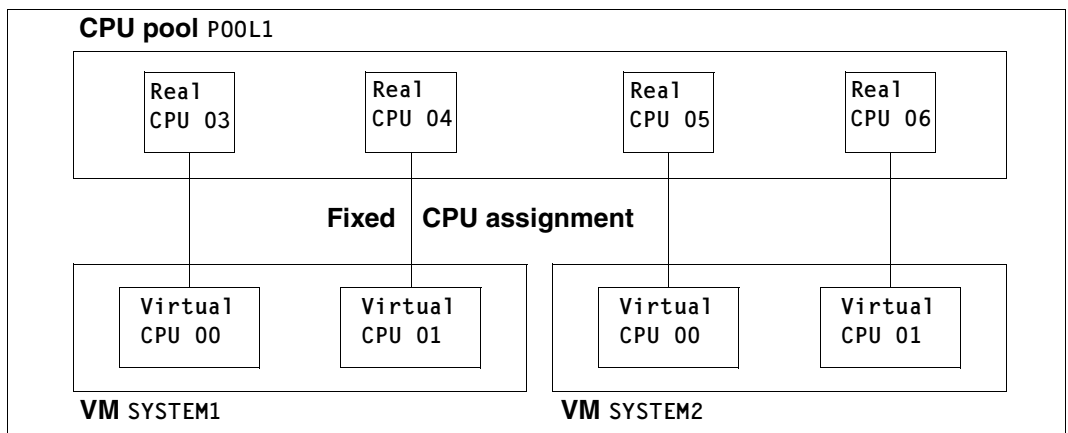


Figure 20: Example of fixed CPU assignment

In the example above, the CPU capacity of `POOL1` on an octo-server would be 50% (4 of 8 CPUs). `SYSTEM1` and `SYSTEM2` each occupy 2 of the 4 CPUs in the pool.

The effective CPU share of each VM is 25% (2 of 8 CPUs of the server), provided the maximum CPU utilization of the VM does not restrict this further.

6.1.4.2 CPU assignment in the time-slicing procedure (S servers)

Normally the number of attached real CPUs in a CPU pool is **less than** the sum of the attached virtual CPUs of all active VMs (RUNNING status) which are assigned to this CPU pool. In this case the hypervisor starts a virtual CPU on a real CPU from the pool using time slicing.

In the VM2000 information commands this scheduling procedure is identified with TS (time slicing).

This CPU assignment makes optimum use of the real CPUs available if direct CPU assignment is not possible.

Operable virtual CPUs of all VMs of the same CPU pool wait for CPU assignment by the VM2000 hypervisor.

If the number of attached real CPUs compared to the number of virtual CPUs increases to the required number (e.g. through reconfiguration or shutdown of a VM), VM2000 automatically switches to the procedure for fixed CPU assignment.

In time slicing, CPUs are assigned in two stages:

1. Selection of the VM group or of the VM that does not belong to a VM group
2. Selection of the virtual CPU of a VM of this VM group or of the VM that does not belong to a VM group which is to run on a free real CPU from the assigned CPU pool

The VM group or the VM that does not belong to a VM group is selected according to the following criteria (see also the [section "Planning distribution of the CPU capacity to the VMs" on page 238](#)):

- CPU quota (CPU-QUOTA)
The greater the CPU quota, the sooner the VM group/VM that does not belong to a VM group will be selected.
- CPU time consumed in the immediate past (CPU intensity)
Here it is not the absolute CPU time after the guest systems were started which is evaluated, but the CPU time consumed within a limited period. This CPU intensity of a VM group/VM that does not belong to a VM group is calculated by the hypervisor and is periodically aged. The CPU intensity depends on the load of the VM group/VM that does not belong to a VM group.
- Maximum CPU utilization (MAX-CPU-UTILIZATION)
When the upper limit for CPU utilization by the VM group/VM that does not belong to a VM group is reached, the virtual CPUs of the VM group/VM that does not belong to a VM group are not started until this upper limit is no longer exceeded (through aging of the CPU intensity).

- The VM group is regarded as a unit
If a VM in the VM group does not use up the CPU share intended for it, the other VMs of this VM group are automatically given preference over VMs which do not belong to this VM group with regard to CPU assignment. In this case the CPU intensity of the VM group is of greater significance than the CPU intensity of the member VM (“load balancing within a VM group”).

CPU affinity

With time slicing, VM2000 uses scheduling to ensure that a virtual CPU runs on the **same** real CPU when the next scheduling operation takes place. This procedure is known as “CPU affinity of the virtual CPU to a real CPU”. It improves the performance of the server under VM2000 during peak operation.

However, the primary goal is still to optimize the response time, i.e.:

- no IDLE state for a real CPU while a virtual CPU of a VM in the CPU pool is ready
- orderly distribution of the CPU capacity of the real CPUs in the CPU pools to the virtual CPUs.

Size of the time slice

The size of the time slice for each VM of VM2000 is defined dynamically in the range 0.1 through 8.0 milliseconds. VMs with a “very small” CPU quota are then also assigned a smaller time slice.

The size of the time slice is recalculated when the VM's effective CPU share is modified (EFF-Q, see [page 243](#)).

Example

Time slice size for a bi-VM with EFF-Q=0,5 on a four-processor server:

$$0,5 * 4 \text{ (real CPUs)} / 2 \text{ (virtual CPUs)} = 1,0 \text{ ms}$$

As of EFF-Q=4,0 the VM would have the existing time slice of 8 ms.

For EFF-Q < 0,05 scheduling every 100 ms is no longer guaranteed.

/SHOW-VM-STATUS INFORMATION=*SCHEDULE outputs the size of the time slice which is currently set for the VM in the CPU-specific information block, output field TIME SLICE, see [page 620](#).

It is the task of the VM2000 administrator to ensure that the effective (EFF-Q, see [page 243](#)) and consequently the current (CUR-Q, see [page 248](#)) CPU share of the VM is large enough to permit the VM to obtain control of the CPU sufficiently frequently, generally once a second.

6.1.5 Example of CPU management

This example explains the various outputs for CPU management in the various VM2000 information commands on S servers.

An example relating specifically to the management of the CPU pool is provided on [page 233](#).

```

/show-vm-resources information=*cpu
VM-ID      CPU-POOL  GR-CPU-Q  CPU-Q/  EFF-Q  MAX-U/  EFF-U  PROC
1 MONITOR  *STDPOOL                20.00/ 10.00 100.00/ 25.00 1(+1)
2 SYSTEM2  *STDPOOL   80.00     70.00/ 28.00 100.00/ 40.00 2(+1)  (1)
3 SYSTEM3  *STDPOOL   80.00     30.00/ 12.00  20.00/ 20.00 1(+1)
4 SYSTEM4  EXCLPOOL                1.00/ 50.00 100.00/ 50.00 2(+1)  (2)
-----
VM-ID      SCHED  ACT-I  VIRT  CPUS
1 MONITOR  DC     NO     00(RUN),01(SLEEP)  (3)
2 SYSTEM2  -     NO     00(WAIT),01(WAIT),02(OFF)
3 SYSTEM3  -     NO     00(WAIT),01(OFF)
4 SYSTEM4  DC    AT-DC  00(RUN),01(RUN),02(OFF)  (4)

CPU-ATTR  CPU-POOL  REAL CPUS
NORMAL    *STDPOOL  00(ATT),01(ATT)  (6)
          EXCLPOOL  02(ATT),03(ATT)
EXTRA     *STDPOOL  04(DET)
SPARE     *STDPOOL  05(DET)

```

- (1) Both a monitor VM with the CPU quota 20 and a VM group `EXTGROUP` with the CPU quota 80 (column `GR-CPU-Q`) have been created for this example. In other words the monitor VM and the VM group `EXTGROUP` are to share the CPU capacity of the standard CPU pool in a ratio of 20:80.

The member CPU quota for the VMs `SYSTEM2` and `SYSTEM3` of the VM group `EXTGROUP` is shown in the output column `CPU-Q`.

See also `/SHOW-VM-GROUP`, (7).

- (2) The VM `SYSTEM4` is assigned to the CPU pool `EXCLPOOL`.

- (3) The displayed planned values effective CPU quota (EFF-Q) and effective maximum CPU utilization (EFF-U) under idealized conditions are calculated using the formulae and constraints in the [section “Planning distribution of the CPU capacity to the VMs” on page 238](#):
- CPU capacity of *STDPPOOL and EXCLPOOL: 50% each (two of four real CPUs each, see (6))
 - For the VMs and VM groups in the standard CPU pool according to the formulae on [page 245](#):
 - Monitor VM: $\text{EFF-Q} = 20 \cdot 50 / 100 = 10\%$ according to the formula (F1a)
 $\text{EFF-U} = 1 \cdot 100 / 4 = 25\%$ according to the constraint (N3)
 - VM group EXCLGROUP: $\text{EFF-Q} = 80 \cdot 50 / 100 = 40\%$ acc. to the formula (F1a)
 $\text{EFF-U} = 40\%$ by means of command
 - SYSTEM2: $\text{EFF-Q} = 70 \cdot 40 / 100 = 28\%$ according to the formula (F1b)
 $\text{EFF-U} = 40\%$ according to the constraint (N1b)
 - SYSTEM3: $\text{EFF-Q} = 30 \cdot 40 / 100 = 12\%$ according to the formula (F1b)
 $\text{EFF-U} = 20\%$ according to the constraint (N1a)
 - For the VM in the CPU pool EXCLPOOL according to the formulae on [page 243](#):
 - SYSTEM4: $\text{EFF-Q} = 1 \cdot 50 / 1 = 50\%$ according to the formula (F1)
 $\text{EFF-U} = 2 \cdot 100 / 4 = 50\%$ according to the constraint (N2) or (N3)
- (4) In the standard CPU pool the monitor VM is using the scheduling procedure with fixed CPU assignment as only one virtual CPU is running but two real attached CPUs are available (see (6)). The virtual spare CPU 01 is operable (SLEEP status) so that it can immediately take over if the only virtual CPU fails. The VMs SYSTEM2 and SYSTEM3 have not yet been started.
- (5) In the CPU pool EXCLPOOL the (only) VM SYSTEM4 is using the scheduling procedure with fixed CPU assignment with two virtual CPUs on two attached real CPUs (see (6)). The CPU quota and maximum CPU utilization are not relevant here; the VM uses the two available CPUs exclusively. The VM2000 administrator has set the attribute VM-ACTIVE-IDLE=*AT-DEDICATED-CPUS for this VM. This means that the VM retains control over a real CPU even if the virtual CPU which runs on it is inactive.
- (6) Two attached real normal CPUs (ATT status) each are contained in the standard CPU pool and in the CPU pool EXCLPOOL. The CPU capacity of *STDPPOOL and EXCLPOOL is 50% each (2 of 4 real normal CPUs of the server, according to the formula on [page 242](#)). The detached extra and spare CPUs (DET status) are not relevant for this example.

```

/show-vm-group vm-group-name=*all,information=*all _____ (7)
VM-GROUP CPU-POOL CPU-Q MAX-U
EXTGROUP *STDPOOL 80.00 40.00

VM-GROUP CPU-POOL VM-ID CPU-Q / EFF-Q MAX-U / EFF-U
EXTGROUP *STDPOOL 2 SYSTEM2 70.00 / 28.00 100.00 / 40.00 _____ (8)
3 SYSTEM3 30.00 / 12.00 20.00 / 20.00

/start-vm ip1-unit=e057,vm-id=system2,information-byte=*dialog _____ (9)

```

- (7) For this example a VM group `EXTGROUP` has been created with a planned CPU quota of 80% and a planned maximum CPU utilization of 40% of the server's CPU capacity (in this example this corresponds to 80% of the CPU capacity of the standard CPU pool).
- (8) The VM group `EXTGROUP` contains two VMs:
- `SYSTEM2` with a member CPU quota 70 and unrestricted CPU utilization
 - `SYSTEM3` with a member CPU quota 30 and maximum CPU utilization of 20.
- In other words `SYSTEM2` and `SYSTEM3` should share the CPU capacity intended for the VM group in a ratio of 70:30.
- (9) The VM2000 administrator starts the guest system on the VM `SYSTEM2`.

```

/show-vm-resources information=*cpu
VM-ID      CPU-POOL  GR-CPU-Q  CPU-Q/  EFF-Q  MAX-U/  EFF-U  PROC
1 MONITOR  *STDPOOL          20.00/ 10.00 100.00/ 25.00 1(+1)
2 SYSTEM2  *STDPOOL    80.00  70.00/ 28.00 100.00/ 40.00 2(+1)
3 SYSTEM3  *STDPOOL    80.00  30.00/ 12.00  20.00/ 20.00 1(+1)
4 SYSTEM4  EXCLPOOL          1.00/ 50.00 100.00/ 50.00 2(+1)

VM-ID      SCHED  ACT-I  VIRT  CPUS
1 MONITOR  DC     NO     00(RUN),01(SLEEP)
2 SYSTEM2  DC     NO     00(RUN),01(STOP),02(OFF) _____ (10)
3 SYSTEM3  -      NO     00(WAIT),01(OFF)
4 SYSTEM4  DC     AT-DC  00(RUN),01(RUN),02(OFF)

CPU-ATTR  CPU-POOL  REAL CPUS
NORMAL    *STDPOOL  00(ATT),01(ATT)
          EXCLPOOL  02(ATT),03(ATT)
EXTRA     *STDPOOL  04(DET)
SPARE     *STDPOOL  05(DET)

/show-vm-status information=*all,period=*recent(seconds=60) _____ (11)
/SHOW-VM-STATUS DATA OUTPUT: (PERIOD= 62.679 SEC)
CPU TIMER INFORMATION FOR REAL CPU(S) _____ (12)
CPU-POOL  REAL      HPV-ACTIVE      HPV-IDLE      ALL-VMS-ACTIVE
          CPU      TIME/  %      TIME/  %      TIME/  %
*STDPOOL  0.091/ 0.07  124.929/99.66  0.338/ 0.27
          00      0.087/ 0.14  62.266/99.34  0.326/ 0.52
          01      0.004/ 0.01  62.663/99.97  0.012/ 0.02
EXCLPOOL  0.066/ 0.05  0.000/ 0.00  125.292/99.95
          02      0.033/ 0.05  0.000/ 0.00  62.646/99.95
          03      0.033/ 0.05  0.000/ 0.00  62.646/99.95

```

(Output continued on the next page)

- (10) SYSTEM2 has been set up as a biprocessor system and initially starts up on the virtual CPU 00 (IPL-CPU, RUN status). The virtual CPU 01 is still in SVP stop (STOP status). Furthermore, VM2000 has also assigned VM the virtual spare CPU 02. It is not operable (OFF status) as the startup phase is running and later two attached virtual normal CPUs will be available for this VM. Scheduling continues with fixed CPU assignment. Two virtual CPUs are now running on two real CPUs in the standard CPU pool.
- (11) To compare the planned values for distributing the CPU capacity to the VMs with the current measured values, the VM2000 administrator outputs the data collection of the last 60 seconds for all VMs.
- (12) Output of the CPU-specific information block for INFORMATION=*TIMER:
- There is little load on the CPUs of the standard CPU pool
 - The CPUs of the CPU pool EXCLPOOL always remain under the control of the VM owing to the attribute VM-ACTIVE-IDLE=*AT-DEDICATED-CPUS

CPU TIMER INFORMATION FOR VM(S) _____ (13)

CPU-POOL					
VM-GROUP	VIRT	CPU-Q	CUR-Q	CUR-U	VM - ACTIVE
VM-ID	CPU				TIME / %
*STDPOOL(P)			50.00	50.00	0.331/ 0.13
1 MONITOR		20.00	25.00	25.00	0.319/ 0.13
	00				0.319/ 0.12
EXTGROUP(G)		80.00	25.00	25.00	0.012/ 0.00
2 SYSTEM2		70.00	25.00	25.00	0.012/ 0.00
	00				0.012/ 0.00
EXCLPOOL(P)			50.00	50.00	125.351/49.99
4 SYSTEM4		1.00	50.00	50.00	125.351/50.00
	00				62.676/24.99
	01				62.675/24.99

CPU COUNTER INFORMATION FOR REAL CPU(S) _____ (14)

CPU-POOL						
REAL	I-O	INT.	INTERCEPT.	EMULATION	SCHEDULE	
CPU	COUNT/	%	COUNT/	%	COUNT/	%
*STDPOOL	00	429/100	1325/ 50	4/100	1325/ 50	
	01	0/ 00	63/ 2	0/ 00	63/ 2	
EXCLPOOL	02	0/ 00	633/ 24	0/ 00	633/ 24	
	03	0/ 00	633/ 24	0/ 00	633/ 24	

CPU COUNTER INFORMATION FOR VM(S)

CPU-POOL						
VIRT	I-O	INT.	INTERCEPT.	EMULATION	SCHEDULE	
CPU	COUNT/	%	COUNT/	%	COUNT/	%
*STDPOOL						
1 MONITOR	00	429/100	1325/ 50	4/100	1325/ 50	
2 SYSTEM2	00	0/ 00	63/ 2	0/ 00	63/ 2	
EXCLPOOL						
4 SYSTEM4	00	0/ 00	633/ 24	0/ 00	633/ 24	
	01	0/ 00	633/ 24	0/ 00	633/ 24	

(Output continued on the next page)

(13) Output of the VM-specific information block for INFORMATION=*TIMER:

- The CUR-Q and CUR-U columns display the calculated current CPU share and the calculated maximum CPU utilization in relation to the active objects in VM2000 operation (see [page 248](#)). These differ from the planned values under idealized conditions (cf. (3)) as only 4 of 6 virtual CPUs and only 3 of 4 VMs are active. Each virtual CPU thus runs on a real CPU of the server.
- Data for the VM SYSTEM3 is not displayed as no virtual CPU is active in this VM.

(14) Output of the information blocks for INFORMATION=*COUNTER

```

SCHEDULE INFORMATION FOR REAL CPU(S) ----- (15)
CPU-POOL   REAL   <-- TIME SLICE  -->  NON-AFFIN SCHED
           CPU     AVG %RUNOUT %SHORT  %IDLE %LOAD
*STDPOOL   00     0.24      1      62  DEDICATED SCHED
           01     0.19      0      0   DEDICATED SCHED
EXCLPOOL   02     VM-ACTIVE-IDLE  DEDICATED SCHED
           03     VM-ACTIVE-IDLE  DEDICATED SCHED

```

```

SCHEDULE INFORMATION FOR VM(S)
CPU-POOL   VIRT <--   TIME SLICE   -->  NON-AFFIN SCHED  %WAIT
VM-ID      CPU  DEF   AVG %RUNOUT %SHORT  %IDLE %LOAD      TIME
*STDPOOL
 1 MONITOR  00  8.00  0.24      1      62  DEDICATED SCHED  --
 2 SYSTEM2  00  8.00  0.19      0      0   DEDICATED SCHED  --
EXCLPOOL
 4 SYSTEM4  00  99.00 VM-ACTIVE-IDLE  DEDICATED SCHED  --
           01          VM-ACTIVE-IDLE  DEDICATED SCHED  --

```

```

% VMS2023 CPU 01 OF VIRTUAL MACHINE (2,SYSTEM2) STARTED ----- (16)
% VMS2023 CPU 02 OF VIRTUAL MACHINE (2,SYSTEM2) STARTED ----- (17)
% VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (2,SYSTEM2),
      CODE 80FFFFFF, CPU 02
% VMS2011 CPU 02 OF VIRTUAL MACHINE (2,SYSTEM2) STOPPED

```

- (15) Output of the information blocks for INFORMATION=*SCHEDULE
- (16) The second virtual CPU 01 of the guest system on the VM SYSTEM2 is attached. Asynchronous messages are marked with % here.
- (17) The virtual spare CPU 02 is briefly started and then stopped again.

/show-vm-status information=*timer,period=*recent(seconds=180) ----- (18)

/SHOW-VM-STATUS DATA OUTPUT: (PERIOD= 193.793 SEC)

CPU TIMER INFORMATION FOR REAL CPU(S)

CPU-POOL	REAL CPU	HPV-ACTIVE TIME/ %	HPV-IDLE TIME/ %	ALL-VMS-ACTIVE TIME/ %
*STDPOOL		1.973/ 0.51	362.707/93.58	22.906/ 5.91
	00	1.245/ 0.64	187.104/96.55	5.444/ 2.81
	01	0.728/ 0.38	175.603/90.61	17.462/ 9.01
EXCLPOOL		0.204/ 0.05	0.000/ 0.00	387.382/99.95
	02	0.103/ 0.05	0.000/ 0.00	193.690/99.95
	03	0.101/ 0.05	0.000/ 0.00	193.692/99.95

CPU TIMER INFORMATION FOR VM(S)

CPU-POOL	VM-GROUP	VIRT CPU	CPU-Q	CUR-Q	CUR-U	VM - ACTIVE TIME / %
*STDPOOL(P)			50.00	50.00		22.905/ 2.95
# 1 MONITOR			20.00	10.00	25.00	1.269/ 0.16 ----- (19)
		00				1.269/ 0.16
# EXTGROUP(G)			80.00	40.00	40.00	21.636/ 2.79
# 2 SYSTEM2			70.00	40.00	40.00	21.636/ 2.79 ----- (20)
		00				13.977/ 1.80
*		01				7.656/ 0.98
*		02				0.003/ 0.00
EXCLPOOL(P)			50.00	50.00		387.340/49.96
4 SYSTEM4			1.00	50.00	50.00	387.340/49.97
		00				193.670/24.98
		01				193.670/24.98

- (18) The VM2000 administrator checks the resultant changes to the time values.
- (19) The values CUR-Q and CUR-U for the monitor VM are now the same as the planned values EFF-Q and EFF-U as SYSTEM2 is active with two virtual normal CPUs. # indicates the changes to the current CPU share (CUR-Q) for the VMs MONITOR and SYSTEM2, and for the VM group EXTGROUP.
- (20) SYSTEM2, as the only active VM of the VM group EXTGROUP, obtains the VM group's share of the server's CPU capacity while SYSTEM3 is not active. * indicates changes which have resulted in corrupted monitored data of ACTIVE times.

```

/show-vm-resources information=*cpu
VM-ID      CPU-POOL  GR-CPU-Q  CPU-Q/  EFF-Q  MAX-U/  EFF-U  PROC
1 MONITOR  *STDPOOL          20.00/ 10.00 100.00/ 25.00 1(+1)
2 SYSTEM2  *STDPOOL    80.00   70.00/ 28.00 100.00/ 40.00 2(+1)
3 SYSTEM3  *STDPOOL    80.00   30.00/ 12.00  20.00/ 20.00 1(+1)
4 SYSTEM4  EXCLPOOL          1.00/ 50.00 100.00/ 50.00 2(+1)

VM-ID      SCHED  ACT-I  VIRT  CPUS
1 MONITOR  TS     NO    00(RUN),01(SLEEP)
2 SYSTEM2  TS     NO    00(RUN),01(RUN),02(OFF)  _____ (21)
3 SYSTEM3  -      NO    00(WAIT),01(OFF)
4 SYSTEM4  DC     AT-DC 00(RUN),01(RUN),02(OFF)

CPU-ATTR   CPU-POOL      REAL CPUS
NORMAL     *STDPOOL      00(ATT),01(ATT)
           EXCLPOOL      02(ATT),03(ATT)
EXTRA      *STDPOOL      04(DET)
SPARE      *STDPOOL      05(DET)

```

```

/assign-vm-group-to-cpu-pool vm-group-name=extgroup,to-cpu-pool=exclpool,
                             cpu-quota=1,attached-vm-cpus=*check-number _____ (22)

```

```

VMS4627 CPU POOL ASSIGNMENT OF VM GROUP 'EXTGROUP' CHANGED FROM
        CPU POOL '*STDPOOL' TO CPU POOL 'EXCLPOOL'

```

```

/show-vm-resources information=*cpu
VM-ID      CPU-POOL  GR-CPU-Q  CPU-Q/  EFF-Q  MAX-U/  EFF-U  PROC  _____ (23)
1 MONITOR  *STDPOOL          20.00/ 25.00 100.00/ 25.00 1(+1)
2 SYSTEM2  EXCLPOOL    1.00   70.00/ 17.50 100.00/ 40.00 2(+1)
3 SYSTEM3  EXCLPOOL    1.00   30.00/  7.50  20.00/ 20.00 1(+1)
4 SYSTEM4  EXCLPOOL          1.00/ 25.00 100.00/ 50.00 2(+1)

```

... (output abbreviated)

- (21) Three virtual CPUs are now running on two real CPUs in the standard CPU pool. Time slicing (TS) is automatically used for scheduling.
- (22) The VM2000 administrator assigns the VM group `EXTGROUP` to the CPU pool `EXCLPOOL`. 1 is selected as the new CPU quota of the VM group, i.e. the VM group and the VM `SYSTEM4` are to share the CPU capacity of the CPU pool `EXCLPOOL`. The command is executed when the number of attached virtual CPUs of every VM in the VM group is less than or equal to the number of attached real CPUs in the CPU pool to which the VM group is to be assigned.
- (23) The changes for all VMs compared to the output above become visible in this output.

/show-vm-group inf=*all _____ (24)

```
VM-GROUP CPU-POOL CPU-Q MAX-U
EXTGROUP EXCLPOOL 1.00 40.00
```

```
VM-GROUP CPU-POOL VM-ID CPU-Q / EFF-Q MAX-U / EFF-U
EXTGROUP EXCLPOOL 2 SYSTEM2 70.00 / 17.50 100.00 / 40.00
3 SYSTEM3 30.00 / 7.50 20.00 / 20.00
```

/show-vm-status information=*timer,period=*recent(seconds=300)

/SHOW-VM-STATUS DATA OUTPUT: (PERIOD= 306.295 SEC)

CPU TIMER INFORMATION FOR REAL CPU(S)

CPU-POOL	REAL CPU		HPV-ACTIVE		HPV-IDLE		ALL-VMS-ACTIVE	
	TIME/	%	TIME/	%	TIME/	%	TIME/	%
**STDPOOL		0.440/	0.07	610.285/	99.62	1.865/	0.30	
	00	0.424/	0.14	304.107/	99.29	1.764/	0.58	
	01	0.016/	0.01	306.178/	99.96	0.101/	0.03	
*EXCLPOOL		0.667/	0.11	571.003/	93.21	40.920/	6.68	
	02	0.328/	0.11	285.509/	93.21	20.458/	6.68	
	03	0.339/	0.11	285.494/	93.21	20.462/	6.68	

CPU TIMER INFORMATION FOR VM(S)

CPU-POOL	VM-GROUP	VIRT CPU	CPU-Q	CUR-Q	CUR-U	VM - ACTIVE	
						TIME /	%
**STDPOOL(P)			50.00	50.00	1.802/	0.14	
# 1 MONITOR			20.00	25.00	1.802/	0.15	
		00			1.802/	0.14	
*EXCLPOOL(P)			50.00	50.00	41.124/	3.35	
# 4 SYSTEM4			1.00	25.00	40.223/	3.28	
		00			20.239/	1.65	
		01			19.984/	1.63	
# EXTGROUP(G)			1.00	25.00	0.901/	0.07	
# 2 SYSTEM2			70.00	25.00	0.901/	0.07	
		00			0.666/	0.05	
		01			0.235/	0.01	

(24) The changes become visible with these commands, too.
See also the explanations for (19) and (20).

6.2 Reconfiguring Channels, controllers and paths (S server)

Hardware units of the channel or controller classes can be reconfigured locally in the monitor system and in the other guest systems. In the monitor system channels and controllers can also be reconfigured globally (i.e. the reconfiguration applies to other guest systems as well).

The BS2000 reconfiguration commands `/ATTACH-DEVICE` and `/DETACH-DEVICE` are used to attach and detach hardware units.

SQ No channels are available on SQ servers. X2000 emulates devices with a virtual input/output path.

VMGIORAL system parameter

The system parameter `VMGIORAL=YES/NO` can be set in the startup parameter service (`SYSOPT-CLASS2` parameter set) in any guest system that is not the monitor system. It can also be modified during BS2000 operation by means of the BS2000 command `/MODIFY-SYSTEM-PARAMETERS`.

`VMGIORAL=YES` permits the global detachment of a hardware unit in the monitor system and in the relevant guest system. `VMGIORAL=NO` is specified, global detachment of a hardware unit is only permitted when the hardware unit has already been detached in the guest system.

SQ Global reconfiguration of hardware units is not available on SQ servers. The system parameter `VMGIORAL` has no effect there.

The global detachment of a channel by means of `FORCE=*UNCONDITIONAL-OFFLINE` is carried out irrespective of the setting for `VMGIORAL`.

The global attachment of a hardware unit is always carried out irrespective of the setting for `VMGIORAL`.

Paths between hardware units (channel controller, channel device, controller device) can be reconfigured locally in the monitor system and in the other guest systems. In the monitor system the paths can also be reconfigured globally (i.e. the reconfiguration applies to other guest systems as well).

The BS2000 reconfiguration commands `/INCLUDE-DEVICE-CONNECTION` and `/REMOVE-DEVICE-CONNECTION` are used to connect and disconnect paths.

The BS2000 command `/SHOW-DEVICE-CONFIGURATION` displays the status of the hardware units and paths in all guest systems.



The explicit reconfiguration of **devices** is always executed locally in the monitor system and in the guest systems and does not have any effect on other guest systems.

Reconfiguration in the monitor system

When reconfiguring (attaching and detaching or connecting and disconnecting) channels, controllers and paths, BS2000 operators can control whether reconfiguration is to be carried out locally in the monitor system or (globally) in all active guest systems as well.

Global reconfiguration is controlled by means of the `SCOPE` operand in the BS2000 reconfiguration commands (see VM2000-specific syntax as of [page 197](#)):

`SCOPE=*OWN-SYSTEM-ONLY`

Carries out reconfiguration locally in the monitor system.

`SCOPE=*VM2000-GLOBAL`

Carries out reconfiguration (globally) in all active guest systems.

Global detachment and disconnection (`/DETACH-DEVICE` and `/REMOVE-DEVICE-CONNECTION`) can be suppressed or permitted in the guest system by means of the `VMGIORAL` system parameter (see [page 262](#)).

Global detachment or disconnection (`/DETACH-DEVICE` or `/REMOVE-DEVICE-CONNECTION`) for channels (although there is an exception described in *Channel reconfiguration* below), controllers and paths is only executed when **all** the guest systems involved can carry out the order.

When a command is executed, the operator is notified by message `NKR0180` of the number of active guest systems and the number of guest systems that are involved in reconfiguration.

Channel reconfiguration

The disconnection of a channel **at the hardware level** (`offline` status) after its successful detachment in the monitor system (`/DETACH-DEVICE`) is executed by VM2000 depending on the status of the channel in all active guest systems and on the `FORCE` operand. The `SCOPE` operand is irrelevant.

`FORCE=*STD / *YES / *NO(...)`

The channel is only disconnected at the hardware level once all the active guest systems have detached it (locally or in global reconfiguration). If this is the case, the status of the channel changes to `offline`. If not, the operator is informed by message `NKR0082`.

FORCE=*UNCONDITIONAL-OFFLINE

The channel is always disconnected at the hardware level, even if not all of the active guest systems have detached it. The status of the channel changes to `offline`.

A channel whose status is `offline` can no longer be addressed or attached by a guest system.

Moreover, a channel whose status is `offline` is not connected at the hardware level (its status does not change to `online`) until it is attached locally or globally in the VM2000 monitor system.

Reconfiguration in the other guest systems

The reconfiguration of hardware units and paths is always carried out locally in the guest systems and has no effect on other guest systems or the monitor system.



Exception

For a VM with the privilege `IO-RESET=*YES`, VM2000 carries out a reset of the channel at the hardware level if this is requested by the guest system (see [page 145](#)).

If `SCOPE=*VM2000-GLOBAL` is specified, it is rejected in a guest system with message NKR0178.

6.3 Memory management and reconfiguration

Memory management of VM2000 includes

- management of the main memory
- reconfiguration of the main memory
- management of global storage (S servers)

Data spaces

The addressing mode for extending the virtual address space is offered under VM2000 for all VMs. In addition to the existing address space (max. 2 Gbytes), further address spaces for data (max. 2 Gbytes each), known as “data spaces”, can be created. Further information on data spaces is provided in the “Executive Macros” manual [15].


6.3.1 Managing main memory

 On S servers, VM2000 manages the server’s real main memory.

VM2000 addresses the available main memory starting at 0. This addressing is known as the **absolute addressing of VM2000**. The absolute address 0 of VM2000 thus refers to the start of the main memory available for VM2000 operation.

 When a VM is initialized on SQ servers, a share of the server’s main memory is made available to the VM, see [page 33](#).

The guest systems of every VM use addressing which is relative to the start of the main memory which is assigned to them. This addressing is known as the **absolute addressing of the VM**. The absolute address 0 of a VM thus refers to the beginning of the VM's main memory.

 VM2000 manages real main memory and domains of up to 1 Tbyte (terabyte; 1 Tbyte = 1024 Gbytes = 1 048 576 Mbytes).

Accounting for main memory

Main memory usage (`MEMORY-SIZE`) is shown in the VM2000 accounting record for the VM. A VM accounting record is written after a main memory reconfiguration has been completed successfully or when the VM is terminated (see [page 386](#)).

6.3.1.1 Managing main memory on S servers

Main memory on S servers begins on a 1-Mbyte boundary and its size is a multiple of 1 Mbyte.

The main memory of the VM2000 hypervisor is situated at the beginning of the host's absolute main memory.

The main memory size of the VM2000 hypervisor is set by means of the VM2000 parameter `HYPervisor-SIZE=*BY-SYSTEM / <integer 4..255>` in the startup parameter service (see [page 53](#)). The setting is retained even if the monitor system is restarted.

The VM2000 hypervisor main memory is followed by the main memory of the monitor VM. Its minimum size is 64 Mbytes. The main memory size of the monitor VM is defined in the configuration file of the monitor VM (`/CREATE-VM`).

This is followed by the main memory of the other guest systems. The main memory size of the guest systems is defined using `/CREATE-VM`.



It is recommended that the remaining main memory (without the VM2000 hypervisor and monitor VM) is distributed according to the expected CPU shares of VM2 through VMn, analogously to the calculations on [page 243](#). If the recommended main memory configuration is used (is dependent on the model), it is not necessary to make differentiated calculations of the main memory size and main memory distribution.

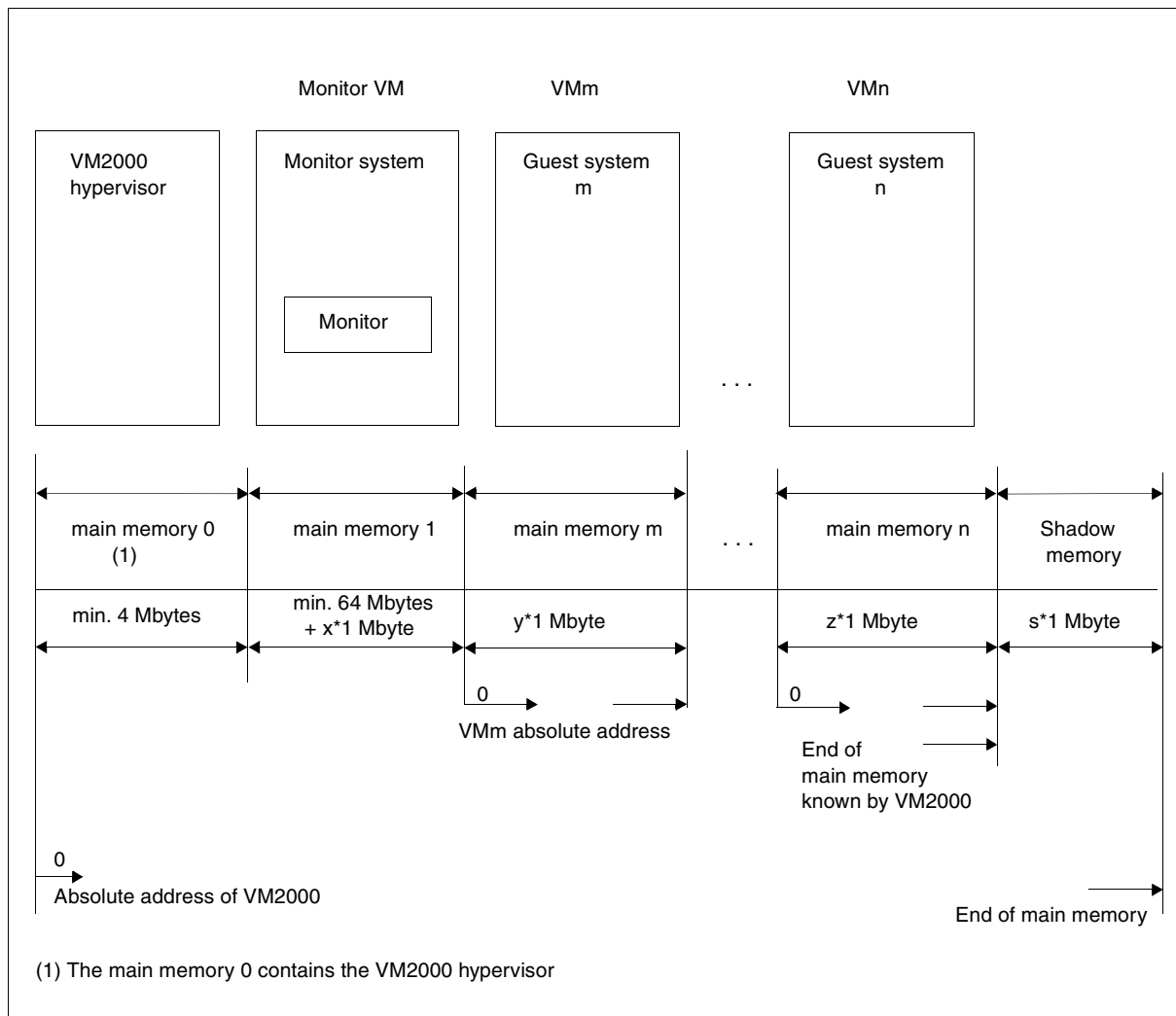


Figure 21: Distribution of main memory (S servers)

Assigning main memory to a VM

The VM2000 administrator determines the size of the main memory for the VM and its position in main memory. This is carried out during initialization of the VM (/CREATE-VM) via the operands MEMORY-SIZE, MIN-MEMORY-SIZE and LOCATION. If the position is not specified, VM2000 selects a suitable area.

The operating system cannot make use of the mainframe's shadow memory (analogous to operation without VM2000). Consequently the amount of available main memory is reduced by the size of the shadow memory. This varies depending on the system configuration used, but is at least 1 Mbyte. The appropriate size can always be determined by reference to the operating instructions.

When assigning free main memory space, VM2000 selects an area with the smallest "wastage", based on the specification in /CREATE-VM. If no suitable area can be found, VM2000 rejects the command.

The VM administrator must then enter a new /CREATE-VM command and attempt to bring the size and location of the main memory into line with the free memory space. He/she can request information on the current distribution of memory space by means of /SHOW-VM-RESOURCES INFORMATION=*MEMORY. It may be necessary to create contiguous memory by means of /MOVE-VM or /REDUCE-VM-MEMORY.

Absolute address decimal (Mbytes)	Main memory	Domain-absolute address decimal (Mbytes)
0	VM2000 hypervisor	
14		0
	Monitor VM	
142		128
	: (Free main memory)	
	.	
256		0
	VM2	
512		256
	: (Free main memory)	
	.	
894		0
	VM3	
1022		128
	Shadow memory	
1024		

Figure 22: Example: Distribution of main memory of 1 Gbytes among three VMs (S servers)

6.3.1.2 Managing main memory on SQ servers

On SQ servers, when a VM is configured it is assigned a share of the main memory which is available in multiples of 2 Mbytes. The main memory of a VM is not contiguous, but scattered.

Consequently the location of a VM in the available main memory need not - as on S servers - be characterized with the start address and the size of the main memory. Nor need a VM on an SQ server be moved with `/MOVE-VM`.

However, the VM's firmware component emulates a contiguous main memory beginning with the address 0 for the BS2000 guest system.



Information on the distribution of the main memory is provided by `/SHOW-VM-RESOURCES INFORMATION=*MEMORY`.

Assigning main memory to a VM

The size of the main memory for the VM is defined by the VM2000 administrator using the `MEMORY-SIZE`, `MIN-MEMORY-SIZE` and `MAX-MEMORY-SIZE` operands when the VM is initialized (`/CREATE-VM`).

A small amount of the main memory of a VM is required for its firmware component. This is visible in the VM2000 information commands as the difference between `MEM-SIZE` and `BS2-MEM-S`, see for example [page 544](#).

The main memory of a VM should be at least 512 Mbytes in size, see [page 140](#).

In addition to the main memory described above, a VM also requires main memory for management data of the Xen hypervisor. This additional main memory is assigned automatically to a VM. It is not displayed in the VM2000 information commands, but reduces the main memory which is available for VM2000 operation and consequently also the largest possible main memory size for a new VM, see [page 598](#).

The size of the main memory of the monitor VM is defined when VM2000 is installed, see [section “Settings for VM2000 in X2000 \(SQ servers\)” on page 62](#). Its minimum size is 512 Mbytes. The standard size of the monitor VM's main memory is 512 Mbytes. This is the recommended value when the monitor VM is used only for VM2000 administration.

Releasing main memory

On SQ servers the release of main memory is executed asynchronously and this can take several seconds. The initialization of a new VM can therefore be delayed.

6.3.2 Reconfiguring main memory

Under VM2000, the main memory of a VM and of the guest system can be expanded or reduced by the VM2000 administrator during the current session (dynamic main memory reconfiguration) to allow a flexible reaction to changes in load or types of tasks without interrupting the operation of the system.

Example

Increasing the size of the main memory of a reserve VM for taking over the load of a failed productive system.

Dynamic reconfiguration of main memory requires measures to be taken in the guest system (if it is active), in the VM's firmware component (SQ server) and in VM2000. It can be carried out for the monitor system as well. Guest systems up to BS2000/OSD V6.0 can only reconfigure main memory up to a size of 32 Gbytes.

The `MEMORY-SIZE` operand is used to dimension the main memory of the VM when the VM is initialized (`/CREATE-VM`). The `MIN-MEMORY-SIZE` and `MAX-MEMORY-SIZE` (SQ server) operands in the same command can be used to specify the minimum and maximum main memory size if the size of the main memory of the VM is altered at a later time. The VM2000 administrator must ensure that the main memory size is not less than the minimum required for running the guest system (see [page 140](#)).

Main memory reconfiguration is possible at all times for inactive guest systems (with the status `INIT-ONLY` or `DOWN`). It is only possible for active guest systems (VM state `RUNNING`) after `SYSTEM READY`. Main memory reconfiguration is not possible for VMs in the `IN HOLD (WAIT)` status.

Any further `/EXTEND-VM-MEMORY` or `/REDUCE-VM-MEMORY` command for the same VM is rejected while a main memory reconfiguration command is being processed. The commands `/SHOW-VM-ATTRIBUTES` and `/SHOW-VM-RESOURCES INFORMATION=*MEMORY/*ALL` provide information on the current, the minimum and the maximum main memory size of the VM and the planned main memory size during main memory reconfiguration.

Since dynamic reconfiguration of main memory requires measures to be taken in the guest system, in the firmware component (SQ server) and in VM2000, the process can be time-consuming. The `/MOVE-VM` (S servers) and `/START-VM` commands can be executed during this time. The new main memory size of the relevant VM is used in the case of `/EXTEND-VM-MEMORY`, and the old main memory size is used in the case of `/REDUCE-VM-MEMORY`. The same applies when the guest system is restarted.

The dynamic reconfiguration of main memory can be monitored with a monitor job variable (see [page 272](#)). Notification of the end of a reconfiguration request is sent to all VM2000 administrators via the message `VMS4094` (successful) or `VMS4095` (only partially successful or unsuccessful) as an asynchronous message, and to the authorized BS2000 consoles or applications via routing code "9". See also [page 131](#).

Termination of main memory configuration

Main memory configuration is terminated in the following cases:

- SETS or shutdown of the guest system
- abortion of the guest system with `/RESET-VM`
- termination of the VM with `/DELETE-VM`
- restarting of the VM with `/START-VM . . . ,CHECK-VM-STATE=*NO`

In all cases, the VM2000 administrator is notified of termination by message VMS4095. This message is also distributed in the monitor system by means of routing code “9”, see also [page 131](#).

In the case of main memory expansion, an accounting record is written for the VM in all cases.

Enlarging main memory

The main memory for a VM can be extended by means `/EXTEND-VM-MEMORY`. At the same time, it is possible to increase the minimum main memory size for the VM.

[SQ] On SQ servers requesting and assigning main memory are not an atomic actions, but take place in several steps. If an error occurs or maximum wait times are exceeded, the main memory of the VM is enlarged only to the size which had been assigned at the time the error occurred.

On SQ servers a VM's main memory can only be enlarged to the maximum main memory size (`MAX-MEMORY-SIZE`).

When the command is processed, actions are first carried out in VM2000 and in the VM's firmware component (SQ server), and then actions are carried out in the guest system (enlarge main memory in the guest system).

Reducing main memory

By means of `/REDUCE-VM-MEMORY`, the main memory for a VM can be reduced up to its minimum main memory size.

[S] On S servers the new main memory size may be below the `MIN-MEMORY-SIZE` of the VM concerned only if the VM is in the `INIT-ONLY` or `DOWN` status. In this case the value for the VM's `MIN-MEMORY-SIZE` is assigned the value of `NEW-MEMORY-SIZE`.

[SQ] On SQ servers the new main memory size may not be below the `MIN-MEMORY-SIZE` of the VM concerned.

When the command is processed, actions are first carried out in the guest system (reduce the main memory in the guest system) and then actions are carried out in VM2000 and in the VM's firmware component (SQ server).



If the main memory of a VM is reduced, the load on the guest system must be reduced accordingly.

Monitoring the reconfiguration of a main memory with job variables (MONJV)

When the reconfiguration of the main memory is initiated by the VM2000 administrator with `/EXTEND-VM-MEMORY` or `/REDUCE-VM-MEMORY`, a monitoring job variable can be defined for this (MONJV). The location and validity of the MONJV correspond to the runtime environment of the sender of the command. VM2000 manages and updates the MONJV as long as the reconfiguration of the main memory is running. By using appropriate procedures, the VM2000 administration can automatically react to the information stored in the MONJV and thus facilitate VM2000 administration.

For general information on job variables under VM2000, see [page 148](#).

For an example of how to use a MONJV to monitor the reconfiguration of a main memory, see [page 274](#).

Beginning and end of monitoring

The MONJV used to monitor the VM is created at the start of the reconfiguration of the main memory with `/EXTEND-VM-MEMORY` or `/REDUCE-VM-MEMORY` by specifying the operand `MONJV`. The monitoring of the reconfiguration of the main memory starts at this point; the MONJV is supplied with data by VM2000.

If a MONJV already exists with the specified name, a password can be specified in the operand `JV-PASSWORD` to check for access authorization.

The monitoring ends with a normal end or an abort of the reconfiguration of the main memory. Even when the monitoring has finished, the MONJV is retained as a user job variable without MONJV protection.

Protecting the MONJV

See the section with this name on [page 149](#).

Location of the MONJV

See the section with this name on [page 149](#).

Values for the MONJV

A MONJV for a VM comprises a system part (bytes 1-128) and a user part (bytes 129-256). Specific values are entered by VM2000 in the system part of the MONJV at specific times as long as the VM exists.

Byte	Meaning / possible values
1-3	Monitoring status: <ul style="list-style-type: none"> – \$S_L : Reconfiguration of main memory started (Started) – \$R_L : Reconfiguration of main memory running (Running) – \$T_L : Reconfiguration of main memory successfully terminated (Terminated) – \$AE : Reconfiguration of main memory aborted with errors (e.g. due to cancellation of running reconfiguration of main memory, see page 271) – \$AF : Reconfiguration of main memory could not be initiated
4	Reserved, has the value zero (0)
5-8	TSN of the job, contains spaces
9-12	Catalog ID of the home pubset of the monitor system
13-16	Reserved, contains spaces
17	Type of MONJV: character V (VM2000)
18-20	Current system sequence number (<integer 1..999>)
21-36	Beginning of monitoring (opening of MONJV) in UTC time (format: yyyy-mm-ddhhmmss)
37-70	Reserved
71-78	Name of VM (<name 1..8>, not for status \$S and \$AF)
79-81	Index of VM printable (<name 3..3>, e.g. 002)
82-86	Memory size of VM in Mbytes (<integer 1..32767>, for \$AE) ¹
87-91	Minimum memory size of VM in Mbytes (<integer 1..32767>, for \$AE) ¹
92-96	Memory size used by the guest system in Mbytes (<integer 1..32767>, for \$AE) ¹
97-106	Memory size of VM in Mbytes (<integer 1..1048576>, for \$AE)
107-116	Minimum memory size of VM in Mbytes (<integer 1..1048576>, for \$AE)
117-126	Memory size used by the guest system in Mbytes (<integer 1..1048576>, for \$AE)
127-128	Reserved
129-256	Reserved for users

Structure of monitoring job variables for reconfiguration of main memory

¹ Specification of the memory size modulo 100 000. The complete values are output from byte 97 onward.

Displaying the MONJV

See the section with this name on [page 151](#).

Behavior when the system is abnormally terminated in the monitor system

MONJVs are retained in the monitor system if the system is terminated abnormally. They have the same status as was set **before** the system was abnormally terminated.



If a VM in the computer network is monitored by a server or a VM outside the monitor VM, the program which is monitoring it must always take the network status of the monitor VM into consideration. HIPLEX MSCF also offers MONJVs to monitor the network status, see the “HIPLEX MSCF” manual [9], especially the section “Monitoring the network status”.

Example

This sample procedure shows how the reconfiguration of a main memory is monitored with MONJV. For an analogous sample procedure using PROP-XT, see [page 133](#).

```

/SET-PROC-OPT
/" *****
/" ** PROCEDURE VMMEM FOR GUEST SYSTEMS WITH VM-ID=2 AND 3 *****
/" *****
/" ** THE PROCEDURE PERFORMS A MEMORY RECONFIGURATION.      "
/" ** IMPORTANT STEPS:                                     "
/" ** - CONNECTION TO VM2000 ESTABLISHED                      "
/" ** - CHECK CURRENT MEMORY SIZE                            "
/" ** - REDUCE MEMORY OF VM2                                "
/" ** - HANDLE ASYNCHRONOUS END VIA MONJV                    "
/" ** - CHECK CURRENT MEMORY SIZE                            "
/" ** - EXTEND MEMORY OF VM3                                  "
/" ** - HANDLE ASYNCHRONOUS END VIA MONJV                    "
/" ** - DISCONNECTION FROM VM2000                            "
/" *****
/" *****
/DECLARE-VARIABLE NAME=SHVAS(TYPE=STRUCTURE),MULT-E=*LIST
/DECLARE-VARIABLE NAME=MIMEMSI(TYP=INT)
/DECLARE-VARIABLE NAME=TXT(TYPE=STRING,INIT-VALUE='OK')
/DECLARE-VARIABLE NAME=JVRES(TYPE=STRING)
/ASSIGN-SYSLST LST.VMMEM
/MODIFY-JOB-OPTIONS LOG=*PAR(LIST=Y)
/BEGIN-BLOCK
/"** CONNECTION TO VM2000  "
/BEGIN-VM-DIALOG VM-ID=1

```

```

/ "*** CHECK CURRENT MEMORY SIZE "
/ EXEC-CMD (SHOW-VM-ATTR VM-ID=2,INF=*MEMORY),STRUCT-OUT=SHVAS
/ IF (SHVAS#.MIN-MEM-SIZE = '*UNDEF')
/ TXT = 'MEMORY CONFIGURATION CURRENTLY NOT POSSIBLE FOR VM-ID=2'
/ GOTO LABEL=ERROR
/ END-IF
/ MIMEMSI=INT(SHVAS#.MIN-MEM-SIZE)
/ IF (SHVAS#.MEM-SIZE = MIMEMSI)
/ TXT = 'CURRENT MEMORY SIZE IS MINIMUM MEMORY SIZE'
/ GOTO LABEL=ERROR
/ END-IF
/ "*** CHECK MEMORY VM-ID=3 "
/ EXEC-CMD (SHOW-VM-ATTR VM-ID=3,INF=*MEMORY),STRUCT-OUT=SHVAS
/ IF (SHVAS#.MIN-MEM-SIZE = '*UNDEF')
/ TXT = 'MEMORY CONFIGURATION CURRENTLY NOT POSSIBLE FOR VM-ID=3'
/ GOTO LABEL=ERROR
/ END-IF
/ "*** REDUCE MEMORY OF VM-ID=2 "
/ REDUCE-VM-MEMORY VM-ID=2,NEW-MEM-SIZE=*MINIMUM,MONJVB=JV.VMMEM.VM02
/ IF-CMD-ERROR
/ TXT = 'ERROR IN REDUCE-VM VM-ID=2'
/ GOTO LABEL=ERROR
/ END-IF
/ "*** SYSTEM MUST WAIT FOR END OF MEMORY CONFIGURATION "
/ WAIT-EVENT *JV(((JV.VMMEM.VM02,1,2) = '$T' OR -
/ (JV.VMMEM.VM02,1,2) = '$A'),TIME-LIM=300)
/ IF-CMD-ERROR
/ TXT = 'ERROR IN WAIT FOR END OF REDUCE-VM VM-ID=2'
/ GOTO LABEL=ERROR
/ END-IF
/ JVBRES = JV('JV.VMMEM.VM02')
/ IF (SUBSTRING(JVBRES,1,2) = '$A')
/ TXT = 'REDUCE-VM VM-ID=2 TERMINATED ABNORMALLY'
/ GOTO LABEL=ERROR
/ END-IF
/ "*** EXTEND MEMORY OF VM3 *****"
/ EXTEND-VM-MEMORY VM-ID=3,NEW-MEM-SIZE=*MAXIMUM,MONJVB=JV.VMMEM.VM03
/ IF-CMD-ERROR
/ TXT = 'ERROR IN EXTEND-VM VM-ID=3'
/ GOTO LABEL=ERROR
/ END-IF

```

```

/ "*** SYSTEM MUST WAIT FOR END OF MEMORY CONFIGURATION  "
/WAIT-EVENT *JV(((JV.VMMEM.VM03,1,2) = '$T' OR -
/          (JV.VMMEM.VM03,1,2) = '$A'),TIME-LIM=300)
/IF-CMD-ERROR
/  TXT = 'ERROR IN WAIT FOR END OF EXTEND-VM VM-ID=3
/  GOTO LABEL=ERROR
/END-IF
/JVRES = JV('JV.VMMEM.VM03')
/IF (SUBSTRING(JVRES,1,2) = '$A')
/  TXT = 'EXTEND-VM VM-ID=3 TERMINATED ABNORMALLY'
/  GOTO LABEL=ERROR
/END-IF
/ "*** REGULAR END"
/END-BLOCK
/IF-BLOCK-ERROR          "HANDLE OTHER ERRORS"
/  TXT = 'OTHER ERRORS'
/  GOTO LABEL = ERROR
/END-IF
/ERROR:
/IF (TXT <> 'OK')
/  SEND-MSG TO=OPER,MSG='*** ERROR IN VMMEM: &TXT ***'
/END-IF
/END-VM-DIALOG 1          "SHUT DOWN CONNECTION TO VM2000"
/IF-CMD-ERROR; END-IF
/DEL-JV JV.VMMEM.VM02
/IF-CMD-ERROR; END-IF
/DEL-JV JV.VMMEM.VM03
/IF-CMD-ERROR; END-IF
/MOD-JOB-OPTIONS LOG=*PAR(LIST=NO)
/ASSIGN-SYSLST *PRIMARY
/EXIT-PROC

```

6.3.3 Global storage management (S server)

Global storage (GS) is semiconductor expanded memory for S servers made nonvolatile by add-ons (battery) (see the “Introductory Guide to System Support” [2]).

SQ Global storage is not available on SQ servers.

Global storage (or the GS complex, see [page 282](#)) can be used in three different ways under VM2000:

- exclusive use of the whole of global storage by a single VM
- shared use of the whole of global storage by several VMs or BS2000/OSD servers
- virtualization of global storage under VM2000. A global storage partition (virtual global storage) can be assigned to a VM for exclusive use or shared use with other VMs of the same server.

Assigning global storage

Each VM can be assigned a **single** global storage partition (virtual) or the whole of global storage (real).

Global storage **should** be assigned to all the VMs before `/START-VM` by means of the VM2000 command `/ADD-VM-RESOURCES`.

Real global storage should be assigned to the monitor VM at startup of the monitor system by means of the configuration file of the monitor VM.

The monitor system and the guest systems can use (real or virtual) global storage exclusively or share it (`VM-SHARED=*NO/*YES`). You can also specify a mode setting for the partitions (`MODE=*MONO/*DUAL`).

Guest systems connect themselves to real or virtual global storage before usage and disconnect themselves after usage.

The outputs for `/SHOW-VM-ATTRIBUTES` and `/SHOW-VM-RESOURCES INFORMATION=*GS/*ALL` display the partition name, attributes and size in Mbytes of the global storage assigned to the VM.

Removing global storage

Global storage is removed with `/DELETE-VM VM-IDENTIFICATION=...` or `/REMOVE-VM-RESOURCES GS=*REAL/*VIRTUAL(...), VM-ID=..., FORCE=*NO/*YES`

Global storage should be removed initially with `FORCE=*NO`. If this is not possible, usage of global storage in the guest system should be terminated. If this is not possible either, global storage must be removed with `FORCE=*YES`.

Exclusive use of real global storage

In this case, the whole of real global storage is assigned to a particular VM. The guest system on this VM uses global storage exclusively. It is possible to write to both GS units (write in duplicate).

Global storage is assigned to the VM as follows:

```
/ADD-VM-RESOURCES GS=*REAL, VM-IDENTIFICATION=..., VM-SHARED=*NO
```

Shared use of real global storage by several VMs

In this case, the whole of real global storage is assigned to several VMs for shared use. It is possible to write to both GS units (write in duplicate).

Global storage is assigned to the VMs as follows:

```
/ADD-VM-RESOURCES GS=*REAL, VM-IDENTIFICATION=..., VM-SHARED=*YES
```

Virtualization of global storage under VM2000

In this case, the whole of **real** global storage is assigned to the monitor VM. In the monitor system, global storage partitions with reserved names are created with BS2000 commands for use as global storage under VM2000.

The VM2000 commands are used to assign and withdraw these partitions as **virtual** global storage to and from the VMs of this server for exclusive or shared use. The virtual global storage assigned appears to the guest system to be a real global storage which can be partitioned as required.

In addition, partitions can be set up in the monitor system for the exclusive use of the monitor system itself (e.g. for DAB).

Global storage cannot be withdrawn from the monitor VM again until there are no longer any global storage partitions assigned to another VM as global storage.

Global storage partitioning and assignment of virtual global storage to the VMs is carried out in three steps:

1. Real global storage is assigned to the monitor VM. This should be done at startup by means of the configuration file of the monitor VM:

```
/ADD-VM-RESOURCES GS=*REAL, VM-IDENTIFICATION=1, VM-SHARED=*NO/*YES
```

2. After SYSTEM READY or after global storage is assigned to the monitor VM, the BS2000 commands for dividing up and using global storage are issued in the monitor system. The partitions for virtualizing global storage under VM2000 are set up as required with the names reserved for VM2000, which are VIRTGS01, VIRTGS02, . . . : VIRTGS16 configured as required:

```
/ATTACH-GS-UNIT . . .
```

```
. . .
```

```
/CREATE-GS-PARTITION PARTITION-ID=VIRTGS01, SIZE=... , MODE=*MONO/*DUAL
```

```
/CREATE-GS-PARTITION PARTITION-ID=VIRTGS02, SIZE=... , MODE=*MONO/*DUAL
```

```
. . .
```

There is a mono-partition in a GS unit.

If there are two GS units, MODE=*DUAL can be used to create partitions consisting of two parts of equal size on the different GS units. This allows data to be stored in duplicate at the hardware level.

The partitions, which are created with /CREATE-GS-PARTITION, are preserved for the next session.

3. After preparations have been made in the monitor system, the VM2000 administrator assigns the partitions to the VMs as virtual global storage. The VM2000 hypervisor then emulates one or two GS units (depending on whether MODE=*MONO or *DUAL is specified) for the guest system. These are of a size corresponding to the partition size in the monitor system.

```
/CREATE-VM VM-NAME=VM2, MEMORY-SIZE=...
```

```
/ADD-VM-RESOURCES GS=*VIRTUAL(PARTITION-ID=VIRTGS01), -
```

```
/ VM-IDENTIFICATION=VM2, VM-SHARED=*NO/*YES
```

```
/CREATE-VM VM-NAME=VM3, MEMORY-SIZE=...
```

```
/ADD-VM-RESOURCES GS=*VIRTUAL(PARTITION-ID=VIRTGS02), -
```

```
/ VM-IDENTIFICATION=VM3, VM-SHARED=*NO/*YES
```

```
. . .
```

The partition retains its mode (mono/dual) at assignment. Guest systems can make use of hardware duplication on dual partitions and thus improve the availability of the data at the hardware level.

To withdraw the global storage partition from a VM and delete global storage partitions, you have to carry out the following three steps:

1. Terminate the use of global storage in the guest system (by terminating the DAB or PUBSET caching by means of `/REDUCE-PAGING-AREA` for paging and `/DETACH-GS-UNIT`).
2. Withdraw the global storage partition (see [page 278](#)).
3. If (virtual) global storage partitions are to be used for something else, you have to withdraw them from the guest system and delete them in the monitor system. You can then create them again as (virtual) global storage partitions (with a different size).

```
/DELETE-GS-PARTITION PARTITION-ID=VIRTGSxx bzw.
/FORCE-DESTROY-GS-PARTITION PARTITION-ID=VIRTGSxx
```

Notes on the use of global storage under VM2000

Changing global storage attributes

Global storage attributes can also be changed while a guest system is running on the VM. You can use the `/REMOVE-VM-RESOURCES` and `/ADD-VM-RESOURCES` commands to change:

- the usage type (virtual GS <---> real GS)
- the mode (mono-partition <---> dual partition)
- the type of operation (VM-SHARED=*NO <---> VM-SHARED=*YES)

The new administration data is detected automatically in the guest system.

Changing the size of global storage

You can change the size of a VM's virtual global storage during guest system operation by proceeding as follows:

- `/DETACH-GS-UNIT GS-UNIT=...` in the guest system
- `/REMOVE-VM-RESOURCES VM-ID=..., GS=*VIRTUAL(PARTITION-ID=VIRTGSxx)`
- In the monitor system: Change the size of the partition (or select a new partition `VM-RESOURCES`, possibly by copying the data from `VM-RESOURCES` to `VM-RESOURCES` with `/COPY-GS-PARTITION`)
- `/ADD-VM-RESOURCES VM-ID=..., GS=*VIRTUAL(PARTITION-ID=<VIRTGSxx or VIRTGSyy>), VM-SHARED=<as in the case of the previous assignment>`
- `/ATTACH-GS-UNIT GS-UNIT=...` in the guest system

The guest system then uses the changed amount of global storage.

For the guest systems, the administration data is adapted automatically, i.e. partitions located entirely within the available address space are preserved.

Backup for guest systems with virtual global storage

To enable guest systems with virtual global storage to be taken over in the event of an error, a backup guest system is set up.

A backup guest system updates its administration data automatically in the event of a takeover.

Accounting for global storage

Use of global storage is recorded in the VM2000 accounting record for the VM. A VM2000 accounting record for the VM (see [page 386](#)) is written after global storage has been assigned successfully, after the assignment has been canceled, when the VM is terminated or after a change in the GS complex configuration.

Global storage reconfiguration

The global storage reconfiguration consists of attaching and detaching global storage (GS) units. Detaching is performed implicitly if a GS unit fails and explicitly with the `/DETACH-GS-UNIT` command. Attaching is performed at system startup or with the `/ATTACH-GS-UNIT` command. `/SHOW-GS-STATUS SELECT=*GS-UNITS` displays the available GS units and their status after system startup.

`/DETACH-GS-UNIT` in the monitor system is only executed if none of the mono partitions on the GS unit concerned is used. If a dual partition of a VM is assigned as virtual global storage, the virtual GS unit on the GS unit to be reconfigured must previously have been detached from the guest system with `/DETACH-GS-UNIT`. It is not necessary to remove the entire virtual global storage of the VM (`/REMOVE-VM-RESOURCES`). The data of the remaining dual partition can be processed further but not duplicated.

After attachment of a GS unit with `/ATTACH-GS-UNIT` in the monitor system, the associated virtual GS unit in the guest system can also be attached with `/ATTACH-GS-UNIT`. At the same time the data of the dual partition is adjusted in the guest system. `/ATTACH-GS-UNIT` in the guest system is only possible if the GS unit in the monitor system is already attached.

GS complex configuration

Global storage (GS) can be shared by several servers of the same type. A cluster of hardware like this is referred to as a **GS complex**. The servers are known as **GS servers**. You will find a detailed treatment of this topic in the “Introductory Guide to Systems Support” [2].

If a GS unit has a connection to a GS server, the GS unit has the status `ATTACHED` and the GS server the status `CONNECTED`. If not, the GS unit has the status `DETACHED` and the GS server the status `DISCONNECTED`. A connection between the GS unit and a GS server is the prerequisite for data transfer between them.

The current GS complex configuration can be displayed by means of `/SHOW-GS-COMPLEX-CONFIGURATION`. The information to be displayed can be selected by means of the `SELECT` operand. If `SELECT=*OBJECTS-OF-OWN-GS-COMPLEX` is specified, the `VM-ID` and host name may also be displayed.

If VM2000 is running on a GS server with several guest systems, global storage is assigned to the VMs by means of `/ADD-VM-RESOURCES` and removed by means of `/REMOVE-VM-RESOURCES` (see [page 277](#)).

If the whole of real global storage is assigned to several VMs on the GS server for sharing (see [page 278](#)), these guest systems work in the same way on the real GS complex. If, in this case, one of the guest systems carries out a GS complex reconfiguration (`/CREATE-GS-COMPLEX` or `/MODIFY-GS-COMPLEX`) or GS server reconfiguration (`/CONNECT-GS-SERVER` or `/DISCONNECT-GS-SERVER`), this affects all of the guest systems involved.



When real global storage is assigned to a VM on a (GS) server that does not (yet) belong to a GS complex, the guest system can initially only execute the commands for GS complex reconfiguration. Global storage cannot be accessed until a GS unit is attached.

In the case of virtual global storage (see [page 278](#)), VM2000 emulates a virtual GS complex. After `/ADD-VM-RESOURCES`, the VM acts as the GS server for this GS complex. For virtual GS complexes, the BS2000 commands for GS unit reconfiguration and GS server reconfiguration can be executed in the relevant guest systems, but this does **not** apply to the commands for GS complex reconfiguration.



- Prerequisites for the assignment of virtual global storage:
- The real GS server must have the status `CONNECTED`.
 - The real GS units affected must have the status `ATTACHED`.

Parallel HIPLEX

A GS complex also offers a communication function via global storage (GSIGP, GS signal processing) that is used by BS2000/OSD in **parallel HIPLEX** for the rapid exchange of global locks. For the shared use of global storage by several BS2000/OSD systems, these must form an XCS cluster. This hardware and software cluster is referred to as “parallel HIPLEX”.

Under VM2000 the servers of a parallel HIPLEX are the guest systems connected via global storage. Up to 15 systems can be configured in a parallel HIPLEX. You will find detailed information on HIPLEX in the “HIPLEX MSCF” manual [9].

6.4 Peripheral device management

The same device peripherals can be operated in native mode and under VM2000. VM2000, the monitor system and all of the guest systems know and manage the same device peripherals.

Information on generating, configuring and determining device peripherals is provided in the [section “Generating and configuring the device peripherals” on page 45](#).

Devices and pubsets

From the set of available devices, VM2000 assigns individual devices or the devices of a pubset to the virtual machines.

The VM2000 administrator obtains detailed information on the device properties and on the assignment of devices to the VMs using `/SHOW-VM-DEVICE-STATUS`. The VM administrator obtains information on the devices of his/her VM using `/SHOW-VM-ATTRIBUTES`.

Devices

Detailed information on devices and their management is provided in the “Device management” chapter of the “Introductory Guide to Systems Support” [2].

Information on managing individual devices under VM2000 is provided in the [section “Assigning and removing devices of a VM” on page 285](#).

Pubsets

Pubsets (public volume sets) are sets of public disks which, in addition to the private volumes, are the storage locations for files in BS2000/OSD. Pubsets contain not only the actual files, but also all the metadata required for file management (e.g. device configuration, file catalog, user catalog). Pubsets are distinguished by their categories (e.g. home, user, shared pubset) and their types (SF and SM pubset).

Detailed information on pubsets and their management is provided in the “Pubset-Management” chapter of the “Introductory Guide to Systems Support” [2].

The product HIPLEX MSCF is required for shared pubsets and XCS pubsets, see the “HIPLEX MSCF” manual [9].

In the device management and information commands of VM2000 pubsets can be addressed by means of their catalog ID or their pubset device. VM2000 determines the devices which belong to the pubset and executes the command for all devices of the pubset, see the [section “Assigning and removing pubsets of a VM” on page 305](#). When the pubset devices are assigned to a VM in this way, VM2000 reacts automatically to changes in the pubset configuration, adapting the assignment and usage mode and the attributes of the devices concerned as needed.

The VM2000 administrator can also manage the devices of the pubset individually, e.g. assign them to the VMs. Changes in the pubset configuration must then be entered manually by the VM2000 administrator. This procedure is not very user-friendly, but also possible.

6.4.1 Assigning and removing devices of a VM

Once a VM is initialized, individual peripheral devices can be assigned to it.

This section examines individual devices. Information on pubset devices is provided in the [section “Assigning and removing pubsets of a VM” on page 305](#).

Devices can be **assigned** to the VMs and thus to the guest systems:

- by the VM2000 or VM administrator using `ADMIN-PRIVILEGES=*MAX` (explicit assignment, see [page 287](#))
- by a guest system whose VM has the privilege `ASSIGN-BY-GUEST=*YES` for the relevant assignment set to which the device is assigned (implicit assignment, see [page 288](#))

In the case of explicit device assignment, the VM2000 or VM administrator specifies a VM's set of devices. Implicit device assignment permits guest system operators to provide the required (free) devices themselves without having to call on an administrator. Both types of device assignment can be used in parallel.

The monitor VM must already have been assigned the devices required for the startup before the monitor system is started up.

S For S servers this is implemented in the configuration file for the monitor VM

SQ For SQ servers this is implemented via the SQ manager (startup settings for the monitor VM)

Devices can be **released** from the VMs and the guest systems

- by the VM2000 or VM administrator (explicit removal using `/REMOVE-VM-DEVICES` or `/SWITCH-VM-DEVICES`, see [page 287](#))
- by a guest system (implicit release of implicitly assigned devices using the BS2000 command `/DETACH-DEVICE`, see [page 290](#))

Device names

Each device is referred to in the VM2000 commands by its **mnemonic device name**, analogously to the BS2000/OSD operating system. The mnemonic device name can be 2-digit (alphanumeric) or 4-digit (hexadecimal).

Access to a device

A guest system cannot access a device until the device is assigned to the VM (implicitly, explicitly, see above) and attached in the guest system (see [page 287](#)).

Accounting for devices

The use of devices is displayed in the VM2000 accounting record for devices (see [page 384](#)).


6.4.1.1 Explicit device assignment and removal

Explicit device assignment

Devices can be assigned explicitly to the VMs and thus the guest systems:

- by the VM2000 administrator, who assigns the required devices to the VMs by using `/ADD-VM-DEVICES` or `/SWITCH-VM-DEVICES` and specifying their mnemonic device names
- by the VM administrator with `ADMIN-PRIVILEGES=*MAX`, who assigns the required devices to his or her VM by using `/ADD-VM-DEVICES` and specifying their mnemonic device names

When the VM is assigned a device by the **VM2000** administrator, the VM administrator receives an information message about every assigned device.

 Alias devices (PAV) cannot be assigned explicitly.

After devices are explicitly assigned to the VM, they have to be attached in the guest system either by means of `/ATTACH-DEVICE` in the guest system or at startup of the guest system (for devices that are generated “attached”). The devices cannot be used until they are attached.


Explicit removal of devices

`/REMOVE-VM-DEVICES` and `/SWITCH-VM-DEVICES` explicitly remove devices from the VM. Both implicitly and explicitly assigned devices can be removed. The devices removed from a VM are no longer available in the associated guest system and should be detached there by means of `/DETACH-DEVICE` before being removed.

The inadvertent removal of **all** of the devices of a VM that is still active can be prevented by means of the `CHECK-VM-STATE` operand (`UNITS=*ALL(CHECK-VM-STATE=*YES/*NO)`).

The removal of devices that are still attached in the guest system is controlled by the `FORCE` operand. The `WAIT-FOR-UNASSIGN` operand can be used to convert explicit device assignment into implicit device assignment. The device then remains with the VM until it is detached in the guest system.

When the **VM2000** administrator removes a device from the VM, the VM administrator receives an information message about every removed device.

 The home pubset and other imported pubsets, paging disks or the last BS2000 console cannot be removed from the monitor VM.

6.4.1.2 Assignment sets, implicit device assignment and release

Rather than explicit device assignment, which is carried out by the VM2000 or VM administrator, it is also possible for guest system operators to assign free devices themselves (implicitly) provided the prerequisites for the VM and the device specified below are met.

Assignment sets

Assignment sets are groups of devices with the `ASSIGN-BY-GUEST` attribute which, for organizational reasons, are regarded as a unit for implicit device assignment to VMs. VMs can consequently also be separated from each other when devices are assigned implicitly.

Assignment sets are created using `/CREATE-VM-ASSIGNMENT-SET` and are assigned a unique name (`SET-NAME`). The assignment set `*STDSET` is created when VM2000 operation is initialized.

Information on the assignment sets which have been created and the VMs to which they are assigned can be obtained using `/SHOW-VM-ASSIGNMENT-SET`. The devices which are assigned to an assignment set can be ascertained using `/SHOW-VM-DEVICE-STATUS`.

Assignment sets are canceled again using `/DELETE-VM-ASSIGNMENT-SET`. The assignment set `*STDSET` cannot be deleted.

Requirements of the device for implicit device assignment

Each device which is envisaged for implicit assignment to a VM must have the `ASSIGN-BY-GUEST` attribute. Devices are assigned the `ASSIGN-BY-GUEST` attribute using `/MODIFY-VM-DEVICE-ATTRIBUTES` and in the process are assigned precisely to one created assignment set. This command can also be used to assign the `ASSIGN-BY-GUEST` attribute and an assignment set to new devices (dynamic I/O configuration modification, see [page 361](#)).

- Ⓢ The default value for the `ASSIGN-BY-GUEST` attribute can be set in the VM2000 parameter file in the monitor system (see [page 53](#)).

Information on whether a device has the `ASSIGN-BY-GUEST` attribute and the assignment set to which it is assigned can be obtained using `/SHOW-VM-DEVICE-STATUS`.

The `ASSIGN-BY-GUEST` attribute can be withdrawn from the device again using `/MODIFY-VM-DEVICE-ATTRIBUTES`.

Requirements of the VM for implicit device assignment

Each VM which is to be able to implicitly assign itself devices must have the `ASSIGN-BY-GUEST` privilege for the assignment set to which the required device is assigned. This privilege can be set using `/CREATE-VM` or `/MODIFY-VM-ATTRIBUTES`; the assignment sets required are then assigned to the VM.

Information on whether a VM has the `ASSIGN-BY-GUEST` privilege and the assignment sets which are assigned to it can be obtained using `/SHOW-VM-ATTRIBUTES/-RESOURCES INFORMATION=*PRIVILEGES`.

`/MODIFY-VM-ATTRIBUTES` enables you to modify the set of assignment sets which is assigned to a VM or to remove the `ASSIGN-BY-GUEST` privilege from the VM.

Using `/SHOW-VM-DEVICE-STATUS SELECT=*PARAMETERS(ASSIGN-BY-GUEST=*YES(SET-NAME=*BY-GUEST(GUEST=<vm-id>))) (/SHOW-VM-DEVICE-STATUS GUEST=<vm-id> for short)` you can ascertain the devices which a guest system may assign itself implicitly.

The BS2000 command `/SHOW-DEVICE-CONFIGURATION` provides information on implicit device assignment options in the guest system (see the “Commands” manual [12] and page 206).



A BS2000 console, the home pubset and the paging disks activated at startup must be assigned to the VM explicitly before the guest system is started.

Implicit device assignment is not available for devices which should already be attached when the guest system starts up.

Implicit device assignment

The BS2000 command `/ATTACH-DEVICE` assigns the affected devices to the VM implicitly. At the same time the command attaches the devices in the guest system. The BS2000 commands `/UNLOCK-DEVICE` and `/SHOW-DEVICE-CONFIGURATION UNIT=*PUBSET-DEVICES` also use implicit device assignment for the duration of command processing.

Usage mode in the case of implicit device assignment:

- Disk devices are assigned for `SHARED` use. Other devices are assigned for `EXCLUSIVE` use. See section “Usage mode of a device” on page 292.
- Devices that are already assigned exclusively to another VM cannot be assigned implicitly (exception: MTC devices in the `KEEP` state, see page 290).

Implicit release of devices

The BS2000 command `/DETACH-DEVICE` releases the implicitly assigned devices in the guest system; they are removed from the VM.

The following points are worth noting about the implicit release of devices:

- Devices assigned to a VM implicitly are released at shutdown of the guest system or at `/RESET-VM`, `/DELETE-VM` or `START-VM` (normal IPL) for the relevant VM.
- In the following cases, implicitly assigned disk devices remain with the VM for diagnostic reasons:
 - abnormal system termination in the guest system
 - system shutdown in the guest system with automatic SLED, e.g. in the case of automatic restart and the restart option `DUMP=*SLED`
 - `/START-VM . . . ,DIAGNOSTIC-IPL=*YES` for the VM

Exclusive use is changed to shared use. Other, implicitly assigned devices are released.

- For MTC devices that are connected to a tape library system, the implicit device assignment continues to apply after `/DETACH-DEVICE`. Pending interruptions for this device can then be dealt with in the guest system. A device like this takes on the status `KEEP`.

If `/SHOW-VM-DEVICE-STATUS` is specified, `GUEST(KEEP)` appears for this device in the `ASSIGNED-BY` output column.

Another VM can assign devices with the status `KEEP` implicitly. A device with the status `KEEP` can be released explicitly by means of `/REMOVE-VM-DEVICES` or `/SWITCH-VM-DEVICES . . . ,FORCE=*NO`.

- In the event of an automatic restart or a restart via SVP in the monitor system, the devices assigned implicitly to the monitor VM are released before `SYSTEM READY`. Message `VMS5227 (S servers)` resp. message `VMS4009` with Insert `BY IPL (SQ servers)` indicate this for each device that is removed.

6.4.1.3 Interplay between implicit and explicit device assignment

If a device is assigned explicitly, this status is not changed when `/ATTACH-DEVICE` or `/DETACH-DEVICE` is used.

A device that is assigned implicitly can subsequently be assigned explicitly by means of `/ADD-VM-DEVICES`. If the device is subsequently detached by means of `/DETACH-DEVICE` in the guest system, this status does not change.

Implicitly assigned devices also be withdrawn explicitly from the VM by the VM2000 or VM administrator by means of `/REMOVE-VM-DEVICES` or `/SWITCH-VM-DEVICES` (see [page 287](#)). After `/SWITCH-VM-DEVICES` the device is assigned explicitly to the new VM.

Explicit assignment of an attached device (attached in the guest system) can be converted to implicit assignment for guest system by means of `/REMOVE-VM-DEVICES . . . , FORCE=*NO(WAIT-FOR-UNASSIGN=*YES)`. Subsequent detachment by means of `/DETACH-DEVICE` in the guest system releases the device.

6.4.1.4 Usage mode of a device

Exclusive use (usage mode EXCLUSIVE)

A device is used exclusively when it is assigned to only one VM and cannot be assigned to any other.

Exclusive use is said to apply in the following cases:

- The device is assigned to the VM explicitly by means of `/ADD-VM-DEVICES` with the operand `TYPE=*NONE/*VC`.
- The device (unless it is a disk device) is assigned to the VM implicitly.
- Exclusive use is set with `/MODIFY-VM-DEVICE-USAGE`.
- Shared use of disks of a pubset is converted to exclusive use when they are imported if the disks are assigned implicitly and only to this VM and the pubset is imported exclusively to this VM in the guest system (`/IMPORT-PUBSET . . . ,USAGE=*EXCLUSIVE`).
- An alias device is used exclusively if its base device is used exclusively (see [page 347](#)).

`/MODIFY-VM-DEVICE-USAGE USAGE=*SHARED` can be used to set shared use for a device that is being used exclusively by a VM, provided it is the kind of device for which shared use is possible (e.g. a disk device).

Inputs/outputs for exclusively assigned devices take place between the guest system and the device directly.

Shared use (usage mode SHARED)

Shared use describes the situation where a device is assigned to at least one VM and can be assigned to other VMs (see also the [section “Shared disk devices” on page 338](#)).

Shared use is said to apply in the following cases:

- The device is assigned to the VM explicitly by means of `/ADD-VM-DEVICES` with the operand `TYPE=*SD`.
- The disk device is assigned to the VM implicitly.
- Shared use is set with `/MODIFY-VM-DEVICE-USAGE`.
- When a pubset which is used exclusively is exported (`/EXPORT-PUBSET`), exclusive use of the disks of the pubset is converted to shared use if the disks are assigned to the VM implicitly (reversal of the procedure for `/IMPORT-PUBSET . . . ,USAGE=*EXCLUSIVE`).
- An alias device is shared if its base device is also shared (see [page 347](#)).

A device that is assigned to a VM for shared use can be assigned to the VM exclusively by means of `/MODIFY-VM-DEVICE-USAGE USAGE=*EXCLUSIVE`, provided it is not already assigned to any other VMs.

Input/output handling for shared devices

The handling of inputs/outputs for shared devices depends on the number of VMs to which the device is assigned:

Direct I/O

If the device is assigned to only one VM, the inputs/outputs take place between the guest system and the device directly.

- SQ On SQ servers these inputs/outputs are handled by X2000. There is no hypervisor overhead.

In the information commands of VM2000, shared devices with direct I/O are identified as such by `SH(D)`.

Indirect I/O

If the device is assigned to a number of VMs on S servers, the inputs/outputs for the shared device are coordinated.

- S On S servers these inputs/outputs are coordinated by the VM2000 hypervisor. The VM2000 hypervisor also receives the asynchronous input/output results and delivers these to the VM concerned.
- SQ On SQ servers these inputs/outputs are coordinated by X2000. There is no hypervisor overhead.

In the information commands of VM2000, shared devices with indirect I/O are identified as such by `SH(I)` for the usage mode.

6.4.1.5 Example of peripheral device management

This example, which consists of several parts, illustrates the different types of device assignment and release and how the usage mode changes when VM2000 and a guest system are used together.

Guest system operations are marked with “***”.

Explicit device assignment

```

/ create-vm memory-size=128,vm-name=system2,cpu-quota=30 _____ (1)
VMS3050 VIRTUAL MACHINE (2,SYSTEM2) CREATED
VMS4615 VIRTUAL MACHINE (2,SYSTEM2) ASSIGNED TO CPU POOL '*STDPPOOL'
/ add-vm-devices units=(z2,z3,d2),vm-id=system2 _____ (2)
VMS4005 DEVICE 'Z2' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
VMS4005 DEVICE 'Z3' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
VMS4005 DEVICE 'D2' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
/ start-vm ip1-unit=d2,vm-id=system2 _____ (3)
***
*** System initialization of the guest system on VM2
***
% VMS2050 GUEST SYSTEM ON VM (2,SYSTEM2) READY _____ (4)

```

- (1) The VM2000 administrator creates a VM.
- (2) The VM2000 administrator assigns KVP console device pair and the device(s) of the home pubset explicitly to VM2. These devices are required by VM2 for startup and therefore cannot be assigned implicitly.
- (3) VM2 is started.
- (4) The guest system on the VM2 has reached SYSTEM READY. Asynchronous messages are marked with % here.

Implicit device assignment

```
/create-vm-assignment-set set-name=sys2impl _____ (5)
```

```
VMS4201 ASSIGNMENT SET 'SYS2IMPL' CREATED
```

```
/modify-vm-attributes vm-id=system2,  
privileges=*par(assign-by-guest=*yes(set-name=sys2impl)) _____ (6)
```

```
/show-vm-device-status units=(d3,d4,d5,x2,x3) _____ (7)
```

```
UNIT DEV-TYPE PAV S ASSIGN-SET USAGE VM-ID/(#VM) ASSIGNED-BY _____ (8)
```

```
D3 D3435 SH(D) 1 MONITOR VM-ADMIN
```

```
D4 D3435
```

```
D5 D3435
```

```
X2 HNC
```

```
X3 HNC
```

```
modify-vm-device-attributes units=(d3,d4,d5,x2,x3),  
assign-by-guest=*yes(set-name=sys2impl) _____ (9)
```

```
/show-vm-device-status select=*par(  
assign-by-guest=*yes(set-name=sys2impl))
```

```
UNIT DEV-TYPE PAV S ASSIGN-SET USAGE VM-ID/(#VM) ASSIGNED-BY _____ (10)
```

```
D3 D3435 SYS2IMPL SH(D) 1 MONITOR VM-ADMIN
```

```
D4 D3435 SYS2IMPL
```

```
D5 D3435 SYS2IMPL
```

```
X2 HNC SYS2IMPL
```

```
X3 HNC SYS2IMPL
```

- (5) The assignment set `SYS2IMPL` is created.
- (6) The VM2000 administrator assigns VM2 the `ASSIGN-BY-GUEST` privilege for the devices in assignment set `SYS2IMPL`. This enables the guest system operator to implicitly assign free devices to the VM (and also to release them again). This privilege could also have been assigned to the VM with `/CREATE-VM` provided the assignment set `SYS2IMPL` already exists at this time.
- (7) The VM2000 administrator wants to make some devices available to VM2 and uses the VM2000 administrator command to obtain information about the status of the devices first.
- (8) All the relevant devices are free. Disk `D3` has already been assigned explicitly to the monitor VM for shared use with other VMs (`VM-ADMIN`) and therefore has the usage mode `SH(D)` (`SHARED with direct I/O`).
- (9) Implicit device assignment is to be permitted for the devices. The VM2000 administrator therefore assigns the `ASSIGN-BY-GUEST` attribute to the relevant devices and assigns them to the assignment set `SYS2IMPL`.
- (10) All the devices concerned now have the `ASSIGN-BY-GUEST` attribute and are assigned to assignment set `SYS2IMPL`.

```

/show-vm-assignment-set set-name=sys2impl ----- (11)
ASSIGNMENT-SET  #DEVICES  VM-ID
SYS2IMPL        5         2 SYSTEM2
***
*** Guest system operation on VM2:
*** /attach-device unit=(d3,d4,d5,x2,x3) ----- (12)
*** NKRO042 'DEVICE      =D3': ATTACH ACCEPTED
*** NKRO042 'DEVICE      =D4': ATTACH ACCEPTED
*** NKRO042 'DEVICE      =D5': ATTACH ACCEPTED
*** NKRO042 'DEVICE      =X2': ATTACH ACCEPTED
*** NKRO042 'DEVICE      =X3': ATTACH ACCEPTED
*** NKRO040 'DEVICE      =X2' ATTACHED
*** NKRO040 'DEVICE      =X3' ATTACHED
*** NKRO040 'DEVICE      =D3' ATTACHED
*** NKRO040 'DEVICE      =D4' ATTACHED
*** NKRO040 'DEVICE      =D5' ATTACHED
*** NKVD010 DISK 'VM03.0' MOUNTED ON DEVICE 'D3' ----- (13)
*** NKVD010 DISK 'VM04.0' MOUNTED ON DEVICE 'D4'
*** NKVD010 DISK 'VM05.0' MOUNTED ON DEVICE 'D5'
*** NBR0740 COMMAND COMPLETED 'ATTACH-DEVICE';
***          (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
***
% VMS4005 DEVICE 'X2' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
          (BY GUEST) ----- (14)
% VMS4005 DEVICE 'X3' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2) (BY GUEST)
% VMS4005 DEVICE 'D3' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2) (BY GUEST)
% VMS4005 DEVICE 'D4' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2) (BY GUEST)
% VMS4005 DEVICE 'D5' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2) (BY GUEST)

```

- (11) The information output for the assignment set shows the related summary.
- (12) The guest system operator uses the BS2000 command /ATTACH-DEVICE to attach the disk and communication devices. These devices are assigned to VM2 implicitly here.
- (13) Different volumes are detected on the attached disks in the guest system.
- (14) The VM and VM2000 administrators are notified about the implicit device assignment to VM2 by asynchronous messages.


```

/show-vm-device-status units=*std,information=*all,
    select=*par(assign-by-guest=*yes(set-name=sys2impl)) _____ (15)
UNIT DEV-TYPE PAV   S ASSIGN-SET  USAGE VM-ID/(#VM) ASSIGNED-BY
D3   D3435                SYS2IMPL  SH(I)   ( 2)   MIXED _____ (16)
                                     1 MONITOR VM-ADMIN
                                     2 SYSTEM2 GUEST
D4   D3435                SYS2IMPL  SH(D)   2 SYSTEM2 GUEST _____ (17)
D5   D3435                SYS2IMPL  SH(D)   2 SYSTEM2 GUEST
X2   HNC                  SYS2IMPL  EXCL    2 SYSTEM2 GUEST _____ (18)
X3   HNC                  SYS2IMPL  EXCL    2 SYSTEM2 GUEST
/show-vm-attributes information=*all,vm-id=system2 _____ (19)
CPU-POOL(P)
VM-ID      VM-GROUP(G) CPU-Q   MAX-U   PROC   MEM-SIZE  MAX-I/O
  2 SYSTEM2 *STDPOOL(P) 30.00  100.00  1(+1)   128 MB (100)

VM-ID      STATE                GUEST  VERSION ADMIN PRIV
  2 SYSTEM2 RUNNING                BS2-OSD V09.0A NO   AG,AS

VM-ID      CPU-POOL  GR-CPU-Q  CPU-Q/ EFF-Q  MAX-U/ EFF-U  PROC
  2 SYSTEM2 *STDPOOL   30.00/ 25.00  100.00/ 25.00  1(+1)

VM-ID      SCHED ACT-I  VIRT CPUS
  2 SYSTEM2 TS     NO    00(RUN),01(SLEEP)

VM-ID      VM-MONJV
  2 SYSTEM2 NO VM-MONJV ASSIGNED

```

(Output continued on the next page)

- (15) The VM2000 administrator obtains detailed information on the status of the devices. If `INFORMATION=*ALL` is specified for a device which is assigned to multiple VMs for shared use, all of the VMs are output.
- (16) Disk D3 is assigned to the monitor VM and VM2. The usage mode is therefore `SH(I)` (SHARED with indirect I/O).
- (17) In the case of implicit device assignment, the disk devices are assigned in the usage mode `SHARED`. Input/output takes place by means of direct I/O because the devices are initially assigned to only one VM.
- (18) In the case of implicit device assignment, other devices (see [page 289](#)) are assigned in the usage mode `EXCLUSIVE`.
- (19) This command provides a VM administrator with information on his or her VM.

```

VM-ID          UNITS _____ (20)
  2 SYSTEM2    ASSIGNED BY VM-ADMIN
                EXCL:  D2,Z2,Z3
                ASSIGNED BY GUEST
                EXCL:  X2,X3
                SHARED: D3,D4,D5
...
VM-ID          PRIV  DETAILS _____ (21)
  2 SYSTEM2    AG    SYS2IMPL
                AS
...
***
*** Guest system operation on VM2:
*** /import-pubset pubset=vm04,use=*exclusive _____ (22)
*** DMS035B IMPORT PUBSET TASK WITH TSN 'XAAL' FOR PUBSET WITH PUBSET ID
***          'VM04' HAS BEEN CREATED AND STARTED
*** SRM5017 USER GROUP MANAGEMENT ON PUBSET 'VM04' INITIALIZED WITH '1'
***          USER GROUP(S)
*** PRO6000 GUARDS CATALOG ':VM04:$TSOS.SYSCAT.GUARDS' OPENED
*** DMS038F IMPORT PUBSET PROCESSING COMPLETED . PUBSET WITH PUBSET ID
***          'VM04' IN 'LOCAL-IMPORTED' STATE
*** NBR0740 COMMAND COMPLETED 'IMPORT-PUBSET';
***          (RESULT: SC2=000, SC1=000, MC=DMS038F); DATE: <date>
*** /show-device-configuration unit=d4,information=*vm2000 _____ (23)
*** MNEM UN-CLASS UN-TYPE CONF-STATE  VM-ASSIGN  VM-USAGE _____ (24)
*** D4  DVC      D3435  ATTACHED    BY-GUEST   EXCL _____ (25)
*** NBR0740 COMMAND COMPLETED 'SHOW-DEVICE-CONFIGURATION'; (RESULT:
***          SC2=000, SC1=000, MC=CMD0001); DATE: <date>
***

```

- (20) The type of device assignment (BY VM-ADMIN, BY GUEST) and the usage mode (EXCL, SHARED) are displayed for VM2.
- (21) The ASSIGN-BY-GUEST privilege for the VM and the assignment set assigned are displayed.
- (22) In the guest system, pubset VM04 is imported in exclusive access mode. The pubset consists of disk device D4.
- (23) The guest system operator obtains information on the VM2000 status of the implicitly assigned disk devices of his or her guest system.
- (24) The output columns VM-ASSIGN and VM-USAGE also inform the guest system operator about the type of assignment and usage.
- (25) For the disk devices D4, the exclusive importing of the pubset has changed the usage mode from SHARED to EXCLUSIVE.

Changing the usage mode

```

/show-vm-device-status units=*std,information=*all,
                        select=*par(state=*assigned(vm-id=system2)) ----- (26)
UNIT DEV-TYPE PAV   S ASSIGN-SET  USAGE VM-ID/(#VM) ASSIGNED-BY
D2   D3435                EXCL  2 SYSTEM2  VM-ADMIN
D3   D3435                SYS2IMPL SH(I)   ( 2)    MIXED
                                     1 MONITOR VM-ADMIN
                                     2 SYSTEM2 GUEST
D4   D3435                SYS2IMPL EXCL  2 SYSTEM2  GUEST ----- (27)
D5   D3435                SYS2IMPL SH(D)  2 SYSTEM2  GUEST
Z2   SKP2                  EXCL  2 SYSTEM2  VM-ADMIN
Z3   SKP2                  EXCL  2 SYSTEM2  VM-ADMIN
X2   HNC                   SYS2IMPL EXCL  2 SYSTEM2  GUEST
X3   HNC                   SYS2IMPL EXCL  2 SYSTEM2  GUEST
/show-vm-device-usage units=d4,vm-id=system2,usage=*shared ----- (28)
/show-vm-device-status units=d4
UNIT DEV-TYPE PAV   S ASSIGN-SET  USAGE VM-ID/(#VM) ASSIGNED-BY
D4   D34375                SYS2IMPL SH(D)  2 SYSTEM2  GUEST ----- (29)

```

- (26) The VM2000 administrator obtains detailed information on the status of the devices assigned to VM2.
- (27) For the disks D4, the exclusive importing of the pubset has changed the usage mode from SHARED to EXCLUSIVE, see (25).
- (28) The VM2000 administrator nevertheless wants to set SHARED use for this device again.
- (29) In spite of the exclusive use of the pubset, SHARED usage is set again in the guest system.

Implicit device release

```

***
*** Guest system operation on VM2:
*** /export-pubset pubset=vm04 _____ (30)
*** DMS035D EXPORT-PUBSET TASK WITH TSN 'XAAN' FOR PUBSET WITH PUBSET
*** ID 'VM04' HAS BEEN CREATED AND STARTED
*** PRO6005 GUARDS CATALOG ':VM04:$TSOS.SYSCAT.GUARDS' CLOSED
*** DMS037F EXPORT-PUBSET PROCESSING COMPLETED FOR PUBSET 'VM04'
*** NBR0740 COMMAND COMPLETED 'EXPORT-PUBSET';
*** (RESULT: SC2=000, SC1=000, MC=DMS037F); DATE: <date>
*** /detach-device unit=d4 _____ (31)
*** NKR0092 'DEVICE =D4': /DETACH-DEVICE PENDING ACCEPTED
*** NKR0045 *** 'DEVICE =D4' DETACHED ***
*** NBR0740 COMMAND COMPLETED 'DETACH-DEVICE';
*** (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
***
% VMS4009 DEVICE 'D4' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
(BY GUEST) _____ (32)
/show-vm-device-status units=d4,information=*all
UNIT DEV-TYPE PAV S ASSIGN-SET USAGE VM-ID/(#VM) ASSIGNED-BY
D4 D34375 SYS2IMPL _____ (33)

```

- (30) Pubset VM04 is exported in the guest system.
- (31) The pubset device(s) are detached in the guest system and consequently removed from the VM.
- (32) VM and VM2000 administrators are informed of the implicit device release by asynchronous messages.
- (33) The device has its original status again.

Converting from explicit to implicit device assignment

```
/add-vm-devices units=d4,vm-id=system2 _____ (34)
```

```
VMS4005 DEVICE 'D4' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
```

```
/show-vm-device-status units=d4,information=*std
```

```
UNIT DEV-TYPE PAV S ASSIGN-SET USAGE VM-ID/(#VM) ASSIGNED-BY _____ (35)
```

```
D4 D34375 SYS2IMPL EXCL 2 SYSTEM2 VM-ADMIN
```

```
***
```

```
*** Guest system operation on VM2:
```

```
*** /attach-device unit=d4 _____ (36)
```

```
*** NKRO042 'DEVICE =D4': ATTACH ACCEPTED
```

```
*** NKRO040 'DEVICE =D4' ATTACHED
```

```
*** NKVD010 DISK 'VM04.0' MOUNTED ON DEVICE 'D4'
```

```
*** NBRO740 COMMAND COMPLETED 'ATTACH-DEVICE';
```

```
*** (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
```

```
***
```

```
/remove-vm-devices units=d4,vm-id=system2 _____ (37)
```

```
VMS4085 DEVICE 'D4' STILL ATTACHED IN GUEST SYSTEM. DEVICE NOT REMOVED
```

```
/remove-vm-devices units=d4,vm-id=system2,
```

```
force=*no(wait-for-unassign=*yes) _____ (38)
```

```
VMS4087 DEVICE 'D4' REMAINS ASSIGNED TO VM (2,SYSTEM2) UNTIL DETACH
```

```
IN GUEST SYSTEM HAS BEEN DONE
```

- (34) The VM2000 administrator assigns the device(s) of the pubset explicitly and exclusively to VM2.
- (35) The output of /SHOW-VM-DEVICE-STATUS indicates this.
- (36) The device is attached in the guest system.
- (37) The VM2000 administrator cannot deallocate the device because it is still attached in the guest system.
- (38) By means of the **FORCE=*NO(WAIT-FOR-UNASSIGN=*YES)** operand the VM2000 administrator changes explicit assignment to implicit assignment (**FORCE=*YES** should only be used in exceptional cases).

```

/show-vm-device-status units=d4,information=*std
UNIT DEV-TYPE PAV   S ASSIGN-SET  USAGE VM-ID/(#VM) ASSIGNED-BY  (39)
D4   D34375          SYS2IMPL    SH(D)  2 SYSTEM2  GUEST
***
*** Guest system operation on VM2:
*** /detach-device unit=d4  (40)
*** NKRO092 'DEVICE      =D4': /DETACH-DEVICE PENDING ACCEPTED
*** NKRO045 *** 'DEVICE      =D4' DETACHED ***
*** NBR0740 COMMAND COMPLETED 'DETACH-DEVICE';
***          (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
***
%   VMS4009 DEVICE 'D4' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
          (BY GUEST)  (41)
/show-vm-device-status units=d4,information=*std
UNIT DEV-TYPE PAV   S ASSIGN-SET  USAGE VM-ID/(#VM) ASSIGNED-BY  (42)
D4   D34375          SYS2IMPL

```

- (39) The output of /SHOW-VM-DEVICE-STATUS indicates that the device is now assigned implicitly to the VM. At the same time, the usage mode has also changed (SH(D)), see also (17).
- (40) The device(s) are detached in the guest system and thus implicitly released.
- (41) VM and VM2000 administrators are informed of the implicit device release by asynchronous messages.
- (42) The device is free again.

Implicit device release at shutdown

```

/shutdown-vm vm-id=system2,action=*shutdown(quiet-time=0) _____ (43)
VMS2054 SHUTDOWN OF GUEST SYSTEM ON VM (2,SYSTEM2) INITIATED
% VMS2051 GUEST SYSTEM ON VM (2,SYSTEM2) DOWN; REASON: SHUTDOWN
% VMS4009 DEVICE 'D3' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2) _____ (44)
% VMS4009 DEVICE 'D5' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
% VMS4009 DEVICE 'X2' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
% VMS4009 DEVICE 'X3' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
/show-vm-attributes vm-id=system2,information=*all
CPU-POOL(P)
VM-ID      VM-GROUP(G)  CPU-Q   MAX-U   PROC   MEM-SIZE  MAX-IO
  2 SYSTEM2  *STDPOOL(P)  30.00  100.00  1(+1)   128 MB  (100)

VM-ID      STATE          GUEST  VERSION ADMIN PRIV
  2 SYSTEM2  DOWN          BS2-OSD V09.0A  NO    AG,AS _____ (45)

VM-ID      CPU-POOL  GR-CPU-Q  CPU-Q/  EFF-Q  MAX-U/  EFF-U  PROC
  2 SYSTEM2  *STDPOOL          30.00/  25.00  100.00/  25.00  1(+1)

VM-ID      SCHED ACT-I  VIRT CPUS
  2 SYSTEM2  TS    NO    00(IDLE),01(IDLE)

VM-ID      VM-MONJV
  2 SYSTEM2  NO VM-MONJV ASSIGNED

VM-ID      UNITS
  2 SYSTEM2  ASSIGNED BY VM-ADMIN
                EXCL:  D2,Z2,Z3 _____ (46)
...

```

- (43) The VM2000 administrator initiates the shutdown in the guest system.
- (44) The implicitly assigned devices of VM2 are released (asynchronous messages to the VM and VM2000 administrators).
- (45) The concluding information command shows that the guest system on VM2 is no longer active.
- (46) Only the explicitly assigned devices are still with VM2.

Explicit device release upon termination of the VM

```

/delete-vm vm-id=system2 _____ (47)
VMS4009 DEVICE 'Z2' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4009 DEVICE 'Z3' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4009 DEVICE 'D2' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4619 VIRTUAL MACHINE (2,SYSTEM2) REMOVED FROM CPU POOL '*STDPOOL'
VMS4024 VIRTUAL MACHINE (2,SYSTEM2) TERMINATED
/show-vm-assignment-set set-name=sys2impl _____ (48)
ASSIGNMENT-SET #DEVICES VM-ID
SYS2IMPL          5      NONE
/modify-vm-device-attributes units=*by-assignment-set(set-name=sys2impl),
                           assign-by-guest=*no _____ (49)
/show-vm-assignment-set set-name=sys2impl _____ (50)
ASSIGNMENT-SET #DEVICES VM-ID
SYS2IMPL          0      NONE
/delete-assignment-set set-name=sys2impl _____ (51)
VMS4205 ASSIGNMENT SET 'SYS2IMPL' DELETED
/show-vm-assignment-set _____ (52)
ASSIGNMENT-SET #DEVICES VM-ID
*STDSET          0      NONE

```

- (47) VM2 is terminated. The explicitly assigned devices are removed.
- (48) Assignment set SYS2IMPL is no longer assigned to a VM, but it still contains 5 devices.
- (49) Implicit device assignment is no longer to be permitted for the devices in assignment set SYS2IMPL.
- (50) Assignment set SYS2IMPL now no longer contains devices; it can be deleted.
- (51) Assignment set SYS2IMPL is deleted.
- (52) Only the assignment set *STDSET still exists.

6.4.2 Assigning and removing pubsets of a VM

In the device management and information commands of VM2000, pubsets can be addressed by means of their catalog ID (cat-id, identical to the pubset name) or their pubres device.

The operand below determines the pubset in all the VM2000 commands concerned:

```
UNITS = *PUBSET-DEVICES(PUBSET = <cat-id 1..4> /
    *BY-PUBRES-DEVICE(UNIT = <alphanum-name 2..2> / <x-text 4..4> )
```

This operand corresponds to the operand which is used in the BS2000 configuration commands, see the [section “VM2000 in the guest systems” on page 194](#).

VM2000 determines the devices which belong to the pubset and executes the command for all the devices of the pubset. This makes device management simpler because the individual devices of a pubset no longer need to be determined and assigned or removed individually.

In the context of VM2000 device management, the term “pubset” is used for short in this manual when referring to the devices of a pubset. When necessary, the term “pubset devices” is used.



Volume sets **cannot** be addressed in this way using VM2000 commands. Volume sets which, for example, do not currently belong to an SM pubset must be addressed by means of their devices’ mnemonic device names.

The VM2000 administrator can also manage the pubset devices individually. However, individually assigned devices are **not** recognized as a pubset by VM2000 and the adjustments by VM2000 in the case of or after a pubset reconfiguration (see [page 315](#)) are then not available

Pubres device (system disk of a pubset)

In BS2000/OSD the pubres device is the disk device containing the metadata of an SF pubset. For simplicity’s sake, in VM2000 device management a “pubres device” is also understood as the volres device of the control volume set in an SM pubset.

In VM2000 device management the pubres device can assume the PUBRES-FOR-ASSIGNMENT (see [page 307](#)) and PUBRES-FOR-ATTRIBUTES (see [page 308](#)) attributes.

Devices of a pubset

VM2000 determines the pubset devices from the pubset's metadata.

When the pubset is addressed by way of its catalog ID, the pubset and its pubres device must also be entered in the monitor system's MRS catalog.

In order to determine the pubset devices, VM2000 briefly assigns the pubres device of an SF pubset or all volres devices of an SM pubset to the monitor VM. VM2000 then attaches these devices (`/ATTACH-DEVICE`) in order to read the list of pubset devices which is stored in its SVL (Standard Volume Label).

Access to a pubset

Access to a pubset by a guest system is only possible after the corresponding assignment of pubset devices to the VM, after the pubset devices have been attached, and after the pubset in the guest system has been imported.

Home pubset

- S On S servers, the devices of the home pubset are automatically assigned explicitly to the monitor VM at startup of the monitor system.
- SQ On SQ servers, the devices of the home pubset must be assigned explicitly via the SQ manager before the monitor system is started up, see [page 62](#).

The devices of the home pubset cannot be removed from the monitor VM.

Its home set must be assigned explicitly and exclusively to another VM before the guest system is started.

Accounting for pubsets

The use of pubsets is displayed implicitly via the use of the pubset devices in the VM2000 accounting record for devices (see [page 384](#)).

6.4.2.1 Explicit pubset assignment and removal

After a VM has been initialized, this VM can easily be assigned the devices of a pubset (pubset for short) using `/ADD-VM-DEVICES` or `/SWITCH-VM-DEVICES` with the `*PUBSET-DEVICES(...)` operand.

Explicit pubset assignment

Pubsets can be assigned explicitly to the VMs and thus to the guest systems

- by the VM2000 administrator, who assigns the required pubset devices to the VMs by specifying the catalog ID or the pubres device in the `*PUBSET-DEVICES(...)` operand in `/ADD-VM-DEVICES` or `/SWITCH-VM-DEVICES`.
- by the VM administrator with `ADMIN-PRIVILEGES=*MAX`, who assigns the required pubset devices to his or her VM by specifying the catalog ID or the pubres device in the `*PUBSET-DEVICES(...)` operand in `/ADD-VM-DEVICES`.

If explicit pubset assignment has taken place for a VM, the pubres device of the pubset assumes the `PUBRES-FOR-ASSIGNMENT` attribute in VM2000 device management. The attribute therefore always refers to the VM concerned. The attribute is removed again when the pubset or pubres device is explicitly released.

A pubres device (and possibly its DRV mirror disk) with the `PUBRES-FOR-ASSIGNMENT` attribute is marked by the suffix (P) in the `ASSIGNED-BY` column of the information output of `/SHOW-VM-DEVICE-STATUS: VM-ADMIN(P)`.

When the VM is assigned a pubset by the **VM2000** administrator, the VM administrator receives an information message for every assigned pubset device and a termination message for the entire pubset.

After pubsets are explicitly assigned to the VM, the pubset devices have to be attached in the guest system and the pubset has to be imported. This is done using `/ATTACH-DEVICE UNIT=*PUBSET-DEVICES(...)` and `/IMPORT-PUBSET` in the guest system or by starting up the guest system. The pubset cannot be used in the guest system until it has been imported.

Explicit removal of pubsets

`/REMOVE-VM-DEVICES` or `/SWITCH-VM-DEVICES` explicitly removes a pubset which is no longer required from the VM. Before being removed, the pubset should be exported in the guest system using `/EXPORT-PUBSET`; the pubset devices should be detached using `/DETACH-DEVICE`. Implicitly and explicitly assigned pubsets can be removed. The pubset devices removed from a VM are then no longer available in the associated guest system.

The removal of devices that are still attached in the guest system is controlled by the `FORCE` operand. The `WAIT-FOR-UNASSIGN` operand can be used to convert explicit device assignment into implicit device assignment. The pubset devices then remain with the VM until they are detached in the guest system.

When the **VM2000** administrator removes a device from the VM, the VM administrator receives an information message about every released pubset device and a termination message for the entire pubset.

6.4.2.2 Implicit pubset assignment and release

In addition to explicit pubset assignment by the VM2000 or VM administrator, the guest system operator can assign the devices of a pubset (pubset for short) to itself (implicitly) if the requirements described below for the VM and the pubset devices are satisfied.

Requirements of the pubset devices for implicit pubset assignment

All pubset devices of a pubset which is envisaged for implicit assignment to a VM must have the `ASSIGN-BY-GUEST` attribute and should be assigned to precisely one assignment set. Information on assignment sets is provided in the section of the same name on [page 288](#).

The pubset devices assume the `ASSIGN-BY-GUEST` attribute in `/MODIFY-VM-DEVICE-ATTRIBUTES UNITS=*PUBSET-DEVICES(. . .)` and in the process are assigned to a created assignment set.



As a rule, the devices of a pubset should only be assigned to **one** assignment set.

If a pubset was specified using `/MODIFY-VM-DEVICE-ATTRIBUTES UNITS=*PUBSET-DEVICES(. . .)`, the pubres device of the pubset in the VM2000 device management is assigned the `PUBRES-FOR-ATTRIBUTES` attribute. The attribute is withdrawn only if the pubres device was explicitly specified in the device list in `/MODIFY-VM-DEVICE-ATTRIBUTES`. The attribute is retained when `UNITS=*ALL / *PUBSET-DEVICES / *BY-ASSIGNMENT-SET` or a device list with wildcards was specified.

A pubres device (and possibly its DRV mirror disk) with the `PUBRES-FOR-ATTRIBUTES` attribute is marked by the suffix `(P)` in the `ASSIGN-SET` column of the information output of `/SHOW-VM-DEVICE-STATUS`, e.g. `*STDSET(P)`.

Information on whether pubset devices have the `ASSIGN-BY-GUEST` attribute and the assignment set to which they are assigned can be obtained using `/SHOW-VM-DEVICE-STATUS`.

The `ASSIGN-BY-GUEST` attribute can be removed from the device again using `/MODIFY-VM-DEVICE-ATTRIBUTES`.

Requirements of the VM for implicit pubset assignment

Each VM which to be able to implicitly assign itself pubsets must have the `ASSIGN-BY-GUEST` privilege for the relevant assignment sets to which the required pubset devices are assigned. This privilege can be set using `/CREATE-VM` or `/MODIFY-VM-ATTRIBUTES`; the assignment sets required are then assigned to the VM.

Information on whether a VM has the `ASSIGN-BY-GUEST` privilege and the assignment sets which are assigned to it can be obtained using `/SHOW-VM-ATTRIBUTES/-RESOURCES INFORMATION=*PRIVILEGES`.

`/MODIFY-VM-ATTRIBUTES` enables you to modify the set of assignment sets which is assigned to a VM or to remove the `ASSIGN-BY-GUEST` privilege from the VM.

Using `/SHOW-VM-DEVICE-STATUS SELECT=*PARAMETERS(ASSIGN-BY-GUEST=*YES(SET-NAME=*BY-GUEST(GUEST=<vm-id>))) (/SHOW-VM-DEVICE-STATUS GUEST=<vm-id> for short)` you can ascertain the devices which a guest system may assign itself implicitly.

The BS2000 command `/SHOW-DEVICE-CONFIGURATION UNIT=*PUBSET-DEVICES(...)` provides information on the options for implicit pubset assignment to the local VM in the guest system, see the “Commands” manual [12] and [page 206](#).



The home pubset must be assigned explicitly and exclusively to the VM before the guest system is started).

Implicit pubset assignment is not available for pubsets which should already be imported when the guest system starts up.

Implicit pubset assignment

The BS2000 command `/ATTACH-DEVICE UNIT=*PUBSET-DEVICES(...)` assigns all pubset devices to the VM implicitly without a mnemonic device name needing to be specified. At the same time the command attaches the devices in the guest system.

Usage mode in the case of implicit device assignment:

- Pubset devices are assigned for shared use (usage mode `SHARED`).
- Pubset devices which are already assigned exclusively to another VM cannot be assigned implicitly.

Implicit release of pubsets

The BS2000 command `/DETACH-DEVICE UNIT=*PUBSET-DEVICES(...)` releases all devices of the pubset; they are removed from the VM.

The following points are worth noting about the implicit release of devices:

- Pubsets assigned to a VM implicitly are released when the guest system is shut down or when `/RESET-VM`, `/DELETE-VM` or `START-VM` (normal IPL) is issued for the relevant VM.
- In the following cases implicitly assigned pubset devices remain with the VM for diagnostic reasons:
 - abnormal system termination in the guest system
 - system shutdown in the guest system with automatic SLED, e.g. in the case of automatic restart and the restart option `DUMP=*SLED`
 - `/START-VM ... ,DIAGNOSTIC-IPL=*YES` for the VM

Exclusive use is changed to shared use. Other, implicitly assigned devices are released.

6.4.2.3 Interplay between implicit and explicit pubset assignment

If pubset devices are assigned explicitly, this status is not changed when `/ATTACH-DEVICE` or `/DETACH-DEVICE` is used.

A pubset that is assigned implicitly can subsequently be assigned explicitly by means of `/ADD-VM-DEVICES`. If the device is subsequently detached by means of `/DETACH-DEVICE` in the guest system, this status does not change.

Implicitly assigned pubsets can also be removed explicitly from the VM by the VM2000 or VM administrator by means of `/REMOVE-VM-DEVICES` or `/SWITCH-VM-DEVICES`, see [page 307](#). After `/SWITCH-VM-DEVICES` the pubset is assigned explicitly to the new VM.

Explicit assignment of an attached pubset to the guest system can be converted to implicit assignment by means of `/REMOVE-VM-DEVICES ... , FORCE=*NO(WAIT-FOR-UNASSIGN=*YES)`. Subsequent detachment by means of `/DETACH-DEVICE` in the guest system releases the pubset.

In the case of pubset reconfiguration, VM2000 automatically adapts the pubset's assignment mode as required, see the [section "Pubset reconfiguration" on page 315](#).

6.4.2.4 Usage mode of pubsets

Exclusive use (usage mode EXCLUSIVE)

A pubset is used exclusively when it is assigned to only one VM and cannot be assigned to any other.

Exclusive use is said to apply in the following cases:

- The pubset is assigned to the VM explicitly by means of `/ADD-VM-DEVICES UNITS=*PUBSET-DEVICES(...), TYPE=*NONE`.
- Exclusive use is set with `/MODIFY-VM-DEVICE-USAGE UNITS=*PUBSET-DEVICES(...), USAGE=*EXCLUSIVE`.
- Shared use of disks of a pubset is converted to exclusive use when they are imported if the disks are assigned implicitly and only to this VM and the pubset is imported exclusively to this VM in the guest system (`/IMPORT-PUBSET ...,USAGE=*EXCLUSIVE`).

`/MODIFY-VM-DEVICE-USAGE UNITS=*PUBSET-DEVICES(...), USAGE=*SHARED` can be used to set shared use for a pubset (but not for the home pubset) that is used exclusively by a VM.

Inputs/outputs for exclusively assigned devices take place directly between the guest system and the pubset device.

Shared use (usage mode SHARED)

Shared use describes the situation where a pubset is assigned to at least one VM and can be assigned to other VMs, see also the [section “Shared disk devices” on page 338](#).

Shared use is said to apply in the following cases:

- The pubset is assigned to the VM explicitly by means of `/ADD-VM-DEVICES UNITS=*PUBSET-DEVICES(...), TYPE=*SD`.
- The pubset is assigned to the VM implicitly.
- Shared use is set with `/MODIFY-VM-DEVICE-USAGE UNITS=*PUBSET-DEVICES(...), USAGE=*SHARED`
- When a pubset which is used exclusively is exported (`/EXPORT-PUBSET`), exclusive use of the disks of the pubset is converted to shared use if the disks are assigned implicitly to the VM (reversal of the procedure for `/IMPORT-PUBSET... ,USAGE=*EXCLUSIVE`).

A pubset that is assigned to a VM for shared use can be assigned to the VM exclusively by means of `/MODIFY-VM-DEVICE-USAGE UNITS=*PUBSET-DEVICES(...), USAGE=*EXCLUSIVE`, provided it is not already assigned to any other VM.

For information on input/output handling for the devices of shared pubsets, please refer to the section [“Input/output handling for shared devices” on page 293](#).

6.4.2.5 Notes on pubset specification in the VM2000 commands

Certain requirements and constraints must be observed for controlling pubsets in the VM2000 commands:

- **ADD-VM-DEVICES**
 - If the pubres device is specified or can be found by VM2000 by means of the catalog ID but the other pubres devices cannot be found, only the pubres device is assigned to the VM.
 - If another (found) pubres device cannot be assigned to the VM, e.g. because a pubset device is already assigned exclusively to another VM, the entire command is not executed.
 - Pubset devices may already be assigned to the VM. In this case an implicit assignment is changed into an explicit assignment by VM2000. The usage mode is adjusted by VM2000; if this is not possible, the entire command is not executed.
 - When the command is executed successfully, the pubres device is assigned the `PUBRES-FOR-ASSIGNMENT` attribute for this VM. The attribute is removed again in the event of explicit (pubset) release for the VM.
- **REMOVE-VM-DEVICES**
 - If the pubres device is specified or can be found by VM2000 by means of the catalog ID but the other pubres devices cannot be found, only the pubres device is removed from the VM.
 - The `PUBRES-FOR-ASSIGNMENT` attribute is removed from the pubres device for this VM.
 - Pubset devices may already have been removed from the VM.
 - When `FORCE=*NO(WAIT-FOR-UNASSIGN=*NO)` is specified, all pubset devices which are still assigned to the VM must be detached in the guest system; otherwise the entire command is not executed.
 - When `FORCE=*NO(WAIT-FOR-UNASSIGN=*YES)` is specified, the explicit assignment is converted into an implicit assignment by VM2000. In this case the `PUBRES-FOR-ASSIGNMENT` attribute is removed from the pubres device for this VM.
 - When the pubset is removed from the monitor VM, the pubset may not/no longer be imported in the monitor system.

- SWITCH-VM-DEVICES
 - If VM2000 cannot find the pubset devices, the command is not executed.
 - All pubset devices must be assigned to the FROM-VM. No pubset device may be assigned to the TO-VM.
 - If a pubset device cannot be removed and reassigned, the entire command is not executed.
 - When FORCE=*NO is specified, all pubset devices in the guest system of the FROM-VM must be detached.
 - If the FROM-VM is the monitor VM, the pubset may not/no longer be imported in the monitor system.
 - When the command is executed successfully, the pubres device is assigned the PUBRES-FOR-ASSIGNMENT attribute for the TO-VM. The attribute no longer applies after the pubres device has been removed for the FROM-VM.
- MODIFY-VM-DEVICE-ATTRIBUTES
 - If VM2000 cannot find the other pubset devices, the command is executed only for the pubres device.
 - When the command is executed successfully, the pubres device is assigned the PUBRES-FOR-ATTRIBUTES attribute. The attribute is removed only if the pubres device is specified explicitly in the device list in /MODIFY-VM-DEVICE-ATTRIBUTES. The attribute is retained when UNITS=*ALL / *PUBSET-DEVICES / *BY-ASSIGNMENT-SET or a device list with wildcards was specified.
- MODIFY-VM-DEVICE-USAGE
 - If VM2000 cannot find the pubset devices, the entire command is not executed.
 - If the usage mode cannot be changed for a pubset, the entire command is not executed, i.e.:
 - When USAGE=*EXCLUSIVE is specified, no pubset device in usage mode SH(I) may be assigned to the VM.
 - When USAGE=*SHARED is specified, no pubset device may be the current IPL device.
 - All pubset devices must be assigned to the VM.
- SHOW-VM-DEVICE-STATUS
 - If the pubres device is specified or can be found by VM2000 by means of the catalog ID but the other pubset devices cannot be found, only information on the pubres device is output.

6.4.2.6 Pubset reconfiguration

Configuration data and also the disk and volume set configuration of a pubset can be modified via command interfaces of the pubset reconfiguration in BS2000/OSD during pubset operation (dynamic pubset reconfiguration). Basic information on this is provided in the section “Dynamic pubset reconfiguration” in the “Introductory Guide to Systems Support” [2].

Pubset reconfiguration is executed by the guest system for the pubsets which are assigned exclusively to it. For shared pubsets, pubset reconfiguration is initiated by any guest system and executed for all slave sharers under the control of the pubset master.

- VM2000 is informed if the system support of a guest system (as of BS2000/OSD V7.0) uses the BS2000 command `/MODIFY-PUBSET-PROCESSING` during ongoing operation to add (`*ADD`) or remove (`*REMOVE`) volumes to/from a pubset which is assigned to the VM.
- When a pubset is attached in the guest system using `/ATTACH-DEVICE UNIT=*PUBSET-DEVICES(. . .)` in a guest system (as of BS2000/OSD V7.0), VM2000 detects a pubset reconfiguration which was performed by another system. However, only the adding of volumes (`*ADD`) is detected.

In both cases VM2000 adjusts the assignment mode (explicit, implicit), the usage mode (exclusive, shareable) and the device attributes (`ASSIGN-BY-GUEST` attribute) for the pubset devices if necessary, see [page 316](#). In special cases, manual adjustments by the VM2000 administrator may be required, see [page 317](#).

The modified pubset configuration is automatically taken into account in the next VM2000 command which addresses the pubset.

Adjustments by VM2000 when a new pubset device is detected

In the following cases VM2000 implements any adjustments which are required:

- The systems support of a guest system (as of BS2000/OSD V7.0) uses the BS2000 command `/MODIFY-PUBSET-PROCESSING` during ongoing operation to add a new volume (`*ADD`) to a pubset which is assigned to the VM.

In the case of a shared pubset, VM2000 implements the adjustments on the server for all the guest systems involved, provided at least one of these guest systems (as of BS2000/OSD V7.0) is running.

- The systems support of a guest system (as of BS2000/OSD V7.0) uses `/ATTACH-DEVICE UNIT=*PUBSET-DEVICES(...)` to attach a pubset to which new volumes were added from another system beforehand.
- The systems support of a guest system (as of BS2000/OSD V7.0) uses `/ATTACH-DEVICE UNIT=*PUBSET-DEVICES(...)` to attach a pubset whose pubset devices could not (yet) be found by VM2000 during the preceding explicit pubset assignment.

If the pubset is explicitly assigned to the VM, i.e. the pubset device has assumed the `PUBRES-FOR-ASSIGNMENT` attribute for the VM, VM2000 implements the following adjustments:

- If the pubres device has the usage mode `SHARED`, the new pubset device is assigned explicitly to the VM and assumes the usage mode `SHARED`.
- If the pubres device has the usage mode `EXCLUSIVE` and the new pubset device does not have the usage mode `SH(I)`, the new pubset device is assigned explicitly to the VM and assumes the usage mode `EXCLUSIVE`.

If the pubres device and the new pubset device are assigned implicitly to the VM, VM2000 implements the following adjustments:

- If the pubres device has the usage mode `SHARED`, the new pubset device assumes the usage mode `SHARED`.
- If the pubres device has the usage mode `EXCLUSIVE` and the new pubset device does not have the usage mode `SH(I)`, the new pubset device assumes the usage mode `EXCLUSIVE`.

If the device attribute `ASSIGN-BY-GUEST` was assigned for the pubset, i.e. the pubres device has the `PUBRES-FOR-ATTRIBUTES` attribute, or if the default value for the VM2000 parameter `ASSIGN-BY-GUEST` (see [page 491](#)) applies for the device attribute of the pubres device, the new pubset device assumes the same device attribute as the pubres device.

Adjustments by VM2000 when a pubset device is removed

If the systems support of a guest system (as of BS2000/OSD V7.0) uses the BS2000 command `/MODIFY-PUBSET-PROCESSING` during ongoing operation to remove a volume (`*REMOVE`) from a pubset which is assigned to the VM, VM2000 implements the following adjustments, as required:

- If the pubset is assigned explicitly to the VM, i.e. the pubres device has the `PUBRES-FOR-ASSIGNMENT` attribute for the VM, the remote device assumes the assignment mode implicitly.
- If the remote device had the usage mode `EXCLUSIVE`, it assumes the usage mode `SH(D)`.

In the case of a shared pubset, VM2000 implements the adjustments for all guest systems (as of BS2000/OSD V7.0) on the server.

Manual adjustments by the VM2000 administrator

Manual adjustments by the VM2000 administrator are still required in multisystem operation for the special cases listed below as the usage and assignment modes for a device and VM cannot be adjusted by VM2000:

- If the pubset is assigned explicitly to an active VM but is not imported in the guest system and in this status a volume is removed from the pubset by a system on another server using `/MODIFY-PUBSET-PROCESSING`.
- If the home pubset is assigned explicitly to a VM in the `DOWN` status and in this status volumes are added to or removed from the home pubset by another system using `/MODIFY-PUBSET-PROCESSING`.
- If, during pubset reconfiguration, no participating guest system is running on the server (`*ADD`) or not all participating guest systems are running on the server (`*REMOVE`) under BS2000/OSD as of V7.0.

6.4.2.7 Mirror pubsets

Mirror pubsets, in other words pubsets in which the pubset volumes are subject to data mirroring, are supported by VM2000 for assigning and releasing pubsets. The actions of VM2000 depend on the type of data mirroring.

Dual Recording by Volume (DRV)

DRV is a recording method in BS2000/OSD which enables the data to be stored in duplicate on two disks, see the “DRV” manual [5]. Depending on the operating mode, DRV is used to record data of a logical disk on one physical disk (mono mode) or on two physical disks (dual mode). DRV is not involved in conventional SRV mode (Single Recording by Volume).

When a disk is operated in mono mode (or in SRV mode), the required VM2000 function is executed only for the one device when the `*PUBSET-DEVICES` operand is specified for this logical disk. The disk device which is physically available or entered in the MRS catalog must be specified as the pubres device. The device attributes `PUBRES-FOR-ASSIGNMENT` and `PUBRES-FOR-ATTRIBUTES` are displayed for the pubres device when `/SHOW-VM-DEVICE-STATUS` is used, see [page 305](#).

When a disk is operated in dual mode, the required VM2000 function is executed when the `*PUBSET-DEVICES` operand is specified for both devices of the logical DRV disk concerned. Either of the two DRV disk devices can be specified or entered in the MRS catalog as the pubres device. The device attributes `PUBRES-FOR-ASSIGNMENT` and `PUBRES-FOR-ATTRIBUTES` are displayed for both pubres devices when `/SHOW-VM-DEVICE-STATUS` is used, see [page 305](#).

If the DRV configuration changes, e.g. because of a switch from dual to mono mode, the failure of a disk or the resumption of dual mode, this does not immediately result in the pubset device list being changed in the SVL of the pubres device(s). In this case a VM2000 command for the pubset is still effective for the pubset’s “old” device list. The pubset device lists are changed only when one of the pubsets is exported or imported or in the event of dynamic pubset reconfiguration, see [page 315](#).

Mirror disks in the local disk storage system

Clone units and additional mirror units (synonym: Business Continuity Volumes (BCVs)) are additional local mirror disks in external disk storage systems which can be split for other purposes without impairing ongoing input/output operations. See the “SHC-OSD” manual [19].

The following procedure applies for additional mirror units and clone units under VM2000:

- When a pubset with clone units or additional mirror units is operated, the required VM2000 function is executed on the pubset’s original or normal units when the *PUBSET-DEVICES operand is specified, provided the original or normal unit was specified for the pubres device or is entered in the MRS catalog (normal case).
- In exceptional cases one of the pubres device’s clone units or additional mirror units can also be specified. The required VM2000 function is then executed on the set of the pubset’s clone units or additional mirror units which belongs to the specified clone unit or additional mirror unit of the pubres device. In this case the software product SHC-OSD is also required in the monitor system.



The (first) access of SHC-OSD to the configuration data of an external disk storage system can take a few minutes. In this case the VM2000 command is aborted with a timeout (message VMS4252). The VM2000 command should be retried later. The required VM2000 function should be available for the mirror pubset after the configuration data has been read.

To operate via a fibre channel (type FC channel) SHC-OSD requires gatekeeper devices. Gatekeeper devices are devices which are specially configured in the Symmetrix disk storage systems for handling the inputs/outputs of the host components’ SYMAPI functions. In this case one or more gatekeeper devices must therefore be assigned to the monitor VM and be attached in the monitor system.

To operate via a type S channel it is sufficient if any device of the disk storage system is assigned to the monitor VM and is attached in the monitor system.

The device attributes PUBRES-FOR-ASSIGNMENT and PUBRES-FOR-ATTRIBUTES are displayed for the pubres device concerned when /SHOW-VM-DEVICE-STATUS is used, see [page 305](#).



The statements above also apply analogously for local replication functions of other external disk storage systems provided these functions are supported by BS2000/OSD and SHC-OSD.

Mirror disks in remote disk storage systems

When the REC (Remote Equivalent Copy, ETERNUS DX) and Symmetrix Remote Data Facility (SRDF) functions are used, the data of a local disk storage system is mirrored on **remote** disk storage systems, see the "SHC-OSD" manual [19].

When a pubset is this way, the required VM2000 function is executed on all of the pubset's source units when the `*PUBSET-DEVICES` operand is specified. The source unit must be specified as the `pubres` device or entered in the MRS catalog.

Only if VM2000 can access the target units in the remote disk storage system can a target unit also be specified as the `pubres` device. The required VM2000 function is then executed on the corresponding target units of the pubset.

The device attributes `PUBRES-FOR-ASSIGNMENT` and `PUBRES-FOR-ATTRIBUTES` are displayed for the `pubres` device concerned when `/SHOW-VM-DEVICE-STATUS` is used, see [page 305](#).



The statements above also apply analogously for remote replication functions of other external disk storage systems, provided these functions are supported by BS2000/OSD and SHC-OSD.

6.4.2.8 Space provisioning (SPACEPRO) under VM2000

The software product SPACEPRO enables you to monitor the storage saturation level of selected pubsets (SPACEPRO pubsets) and, when required, to extend the number of volumes in the pubset.

The SPACEPRO pubsets are assigned so-called pool pubsets. Pool pubsets consist of volumes which are used to extend SPACEPRO pubsets. When a specific storage saturation level is reached for a SPACEPRO pubset, the SPACEPRO pubset is extended by a volume from the pool pubset assigned.

The SPACEPRO prerequisites and rules must also be observed when the product is used under VM2000. In particular, all the volumes of SPACEPRO and pool pubsets on all the participating (guest) systems must have the same mnemonic device name.

Basic information on space provisioning with SPACEPRO is provided in the “Introductory Guide to Systems Support” [2]. There you will also find information on the use of SPACEPRO under VM2000.

Preparations for space provisioning in VM2000

To permit SPACEPRO to be used effectively in guest systems (BS2000/OSD as of V7.0) under VM2000, the VM2000 administrator must have performed the following preparatory work:

- The SPACEPRO pubsets are assigned explicitly to the VMs or prepared for implicit assignment by the guest system:
 - Explicit pubset assignment (see [page 307](#)) using


```
/ADD-VM-DEVICES UNITS=*PUBSET-DEVICES(PUBSET=<spubvmx>)
```
 - Implicit assignment of pubsets (see [page 309](#)) is prepared for using


```
/CREATE-VM-ASSIGNMENT-SET SET-NAME=<setvmx>
/MODIFY-VM-DEVICE-ATTRIBUTES UNITS=*PUBSET-DEVICES(PUBSET=<spubvmx>),
                               ASSIGN-BY-GUEST=*YES(SET-NAME=<setvmx>)
/MODIFY-VM-ATTRIBUTES VM-ID=<vmx>,
                       PRIVILEGES=*PAR(ASSIGN-BY-GUEST=*YES(SET-NAME=<setvmx>))
```
- The pool pubsets are prepared for implicit assignment by a guest system (see [page 309](#)):


```
/CREATE-VM-ASSIGNMENT-SET SET-NAME=<poolset>
/MODIFY-VM-DEVICE-ATTRIBUTES UNITS=*PUBSET-DEVICES(PUBSET=<poolpub>),
                               ASSIGN-BY-GUEST=*YES(SET-NAME=<poolset>)
```

For all guest systems involved:

```
/MODIFY-VM-ATTRIBUTES VM-ID=<vmx>,
                       PRIVILEGES=*PAR(ASSIGN-BY-GUEST=*YES(*ADD(SET-NAME=<poolset>)))
```

Execution under VM2000

When the preparations for using SPACEPRO in VM2000 have been completed, execution in the guest system does not differ from execution in native mode.

In the example below the SPACEPRO pubset <spubvmx> is imported and in use on the guest system. SPACEPRO has found a storage saturation status of <spubvmx>, and the pubset is to be extended by a volume <poolvol> from the pool pubset <poolpub>. The procedure in the guest system is shown below (executed by SPACEPRO):

```
/ATTACH-DEVICE UNIT=*PUBSET-DEVICES(PUBSET=<poolpub>) _____ (1)
/IMPORT-PUBSET PUBSET=<poolpub>, USAGE=*EXCLUSIVE _____ (2)
/MODIFY-PUBSET-PROCESSING PUBSET=<poolpub>,PUBSET-TYPE=*SINGLE-FEATURE
(VOLUME-ASSIGNMENT=*REMOVE(VOLUME=<poolvol>)) _____ (3)
/START-VOLIN . . . _____ (4)
/MODIFY-PUBSET-PROCESSING PUBSET=<spubvmx>,PUBSET-TYPE=*SINGLE-FEATURE
(VOLUME-ASSIGNMENT=*ADD(VOLUME=<poolvol>)) _____ (5)
/EXPORT-PUBSET PUBSET=<poolpub> _____ (6)
/DETACH-DEVICE UNIT=*PUBSET-DEVICES(pubset=<poolpub>) _____ (7)
```

- (1) SPACEPRO attaches the pool pubset <poolpub>. VM2000 implicitly assigns the pubset devices to the VM in the usage mode SHARED, see [page 309](#).
- (2) SPACEPRO imports the pool pubset <poolpub>. VM2000 assigns the pubset devices the usage mode EXCLUSIVE, see [page 311](#).
- (3) SPACEPRO removes a volume from the pool pubset (pubset reconfiguration). VM2000 changes the usage mode of the device it has removed from EXCLUSIVE to SH(D), see [page 317](#).
- (4) SPACEPRO executes a VOLIN run in order to rename or, if necessary, reformat the removed volume.
- (5) SPACEPRO adds the renamed volume to the SPACEPRO pubset (pubset reconfiguration). VM2000 adapts the assignment and usage modes and the device attribute of the newly added device to the conditions in the SPACEPRO pubset <spubvmx>, see [page 316](#).
- (6) SPACEPRO exports the pool pubset <poolpub>. VM2000 changes the usage mode of the remaining pubset devices from EXCLUSIVE to SH(D).
- (7) SPACEPRO detaches the pool pubset <poolpub>. The pool pubset can now once more be used for space provisioning in other systems. VM2000 implicitly removes the pubset devices from the VM, see [page 310](#). Only the removed device with the volume <poolvol> remains assigned to the VM.

Notes

The modified pubset configuration is automatically taken into account in the next VM2000 command which addresses the pubset.

If the SPACEPRO pubset is a shared pubset, the following adjustments are made to all other sharers of the pubset - also on other servers with VM2000 - in the (guest) systems (as of BS2000/OSD V7.0):

- The newly added device is attached if it has not yet been attached
- The newly added device is reserved (PUBLIC mode)
- The assignment and usage modes and the device attributes of the newly added device are adjusted, see [page 316](#)



If the SPACEPRO pubset is attached but not yet imported in another (guest) system at the time when it is extended, this can result in problems when the pubset is later imported into this (guest) system (in native mode and under VM2000).

If the SPACEPRO pubset is only assigned to another guest system but neither attached nor imported at the time when it is extended, VM2000 detects the pubset reconfiguration when the pubset is attached and adjusts the assignment and usage modes and the device attribute of the newly added device, see [page 316](#).

VM2000 also detects any other pubset reconfigurations, e.g. extension of a pool pubset, and adjusts the assignment and usage modes and the device attributes of the newly added device. If the pubset reconfiguration was performed by another user of the pubset, the change is detected the next time the pubset is attached in a guest system.

6.4.2.9 Example of pubset management

This example, which consists of several parts, illustrates the various ways of assigning and releasing pubsets and how the usage mode is changed when VM2000 and a guest system are used together.

In principle the execution is the same as for [“Example of peripheral device management” on page 294](#), but it utilizes simplified device selection in the command by specifying the pubset.

Guest system operations are marked with “***”.

Explicit pubset assignment

```

/ create-vm memory-size=128,vm-name=system2,cpu-quota=30 ----- (1)
VMS3050 VIRTUAL MACHINE (2,SYSTEM2) CREATED
VMS4615 VIRTUAL MACHINE (2,SYSTEM2) ASSIGNED TO CPU POOL '*STDPPOOL'
/ add-vm-devices units=(z2,z3),vm-id=system2 ----- (2)
VMS4005 DEVICE 'Z2' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
VMS4005 DEVICE 'Z3' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
/ add-vm-devices units=*pubset-devices(pubset=vm02),vm-id=system2
VMS4005 DEVICE 'D2' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
VMS4220 PUBSET 'VM02' WITH PUBRES DEVICE 'D2' ASSIGNED TO VM (2,SYSTEM2)
/ start-vm ip1-unit=d2,vm-id=system2 ----- (3)
***
*** System initialization of the guest system on VM2
***
% VMS2050 GUEST SYSTEM ON VM (2,SYSTEM2) READY ----- (4)

```

- (1) The VM2000 administrator creates a VM.
- (2) The VM2000 administrator explicitly assigns the KVP device pair and the device(s) of the home pubset to VM2. These devices are required by VM2 for startup and therefore cannot be assigned implicitly. Pubset VM02 is entered in the monitor system's MRS catalog and is consequently detected here by VM2000.
- (3) VM2 is started.
- (4) The guest system on the VM2 has reached SYSTEM READY. Asynchronous messages are marked with % here.

Implicit pubset assignment

```

/CREATE-VM-ASSIGNMENT-SET SET-NAME=SYS2IMPL _____ (5)
VMS4201 ASSIGNMENT SET 'SYS2IMPL' CREATED
/MODIFY-VM-ATTRIBUTES VM-ID=SYSTEM2,
      PRIVILEGES=*PAR(ASSIGN-BY-GUEST=*YES(SET-NAME=SYS2IMPL)) _____ (6)
/SHOW-VM-DEVICE-STATUS UNITS=*PUBSET-DEVICES(
      PUBSET=*BY-PUBRES-DEVICE(UNIT=D4)) _____ (7)
% NKRO042 'DEVICE      =D4': ATTACH ACCEPTED
% NKRO040 'DEVICE      =D4' ATTACHED
% NKVD010 DISK 'VM04.0' MOUNTED ON DEVICE 'D4'
% NKRO045 *** 'DEVICE      =D4' DETACHED ***
UNIT DEV-TYPE PAV   S ASSIGN-SET  USAGE VM-ID/(#VM) ASSIGNED-BY
D4   D3435
/SHOW-VM-DEVICE-STATUS UNITS=(D3,D5,X2,X3)
UNIT DEV-TYPE PAV   S ASSIGN-SET  USAGE VM-ID/(#VM) ASSIGNED-BY
D3   D3435                SH(D) 1 MONITOR  VM-ADMIN _____ (8)
D5   D3435
X2   HNC
X3   HNC
/MODIFY-VM-DEVICE-ATTRIBUTES
      UNITS=*PUBSET-DEVICES(PUBSET=*BY-PUBRES-DEVICE(UNIT=D4)),
      ASSIGN-BY-GUEST=*YES(SET-NAME=SYS2IMPL) _____ (9)
VMS4240 DEVICE ATTRIBUTES OF PUBSET 'VM04' WITH PUBRES DEVICE 'D4' MODIFIED
/MODIFY-VM-DEVICE-ATTRIBUTES UNITS=(D3,D5,X2,X3),
      ASSIGN-BY-GUEST=*YES(SET-NAME=SYS2IMPL)

```

- (5) The assignment set `SYS2IMPL` is created.
- (6) The VM2000 administrator assigns VM2 the `ASSIGN-BY-GUEST` privilege for the devices in assignment set `SYS2IMPL`. This enables the guest system operator to implicitly assign free devices to the VM (and also to release them again). This privilege could also have been assigned to the VM with `/CREATE-VM` provided the assignment set `SYS2IMPL` already exists at this time.
- (7) The VM2000 administrator wants to make some devices available to VM2 and uses the VM2000 administrator command to obtain information about the status of the devices first. The `pubres` device `D4` is briefly attached in the monitor system in the course of command execution in order to ascertain the `pubset` data. Similar reconfiguration messages are not shown in the rest of this example.
- (8) All the relevant devices are free. Disk `D3` has already been assigned explicitly to the monitor VM for shared use with other VMs (`VM-ADMIN`) and therefore has the usage mode `SH(D)` (SHARED with direct I/O).
- (9) Implicit device assignment is to be permitted for the devices. The VM2000 administrator therefore specifies the `ASSIGN-BY-GUEST` attribute for the `pubset` and the devices concerned and assigns them to assignment set `SYS2IMPL` zu.

```

/show-vm-device-status select=*par(
                        assign-by-guest=*yes(set-name=sys2impl))
UNIT DEV-TYPE PAV   S ASSIGN-SET  USAGE VM-ID/(#VM) ASSIGNED-BY  (10)
D3   D3435                SYS2IMPL  SH(D)  1 MONITOR  VM-ADMIN
D4   D3435                SYS2IMPL(P)
D5   D3435                SYS2IMPL
X2   HNC                  SYS2IMPL
X3   HNC                  SYS2IMPL
/show-vm-assignment-set set-name=sys2impl  (11)
ASSIGNMENT-SET  #DEVICES  VM-ID
SYS2IMPL        5         2 SYSTEM2

```

- (10) All the devices concerned now have the `ASSIGN-BY-GUEST` attribute and are assigned to assignment set `SYS2IMPL`. The `PUBRES-FOR-ATTRIBUTES` attribute is displayed for device `D4`.
- (11) The information output for the assignment set shows the related summary.

```

***
*** Guest system operation on VM2:
*** /attach-device unit=*pubset-devices(
          pubset=*by-pubres-device(unit=d4)) _____ (12)
*** NKRO042 'DEVICE      =D4': ATTACH ACCEPTED
*** NKRO040 'DEVICE      =D4' ATTACHED
*** NKVD010 DISK 'VM04.0' MOUNTED ON DEVICE 'D4' _____ (13)
*** NBR0740 COMMAND COMPLETED 'ATTACH-DEVICE';
***          (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
*** /attach-device unit=(d3,d5,x2,x3)
*** NKRO042 'DEVICE      =D3': ATTACH ACCEPTED
*** NKRO042 'DEVICE      =D5': ATTACH ACCEPTED
*** NKRO042 'DEVICE      =X2': ATTACH ACCEPTED
*** NKRO042 'DEVICE      =X3': ATTACH ACCEPTED
*** NKRO040 'DEVICE      =X2' ATTACHED
*** NKRO040 'DEVICE      =X3' ATTACHED
*** NKRO040 'DEVICE      =D3' ATTACHED
*** NKRO040 'DEVICE      =D5' ATTACHED
*** NKVD010 DISK 'VM03.0' MOUNTED ON DEVICE 'D3'
*** NKVD010 DISK 'VM05.0' MOUNTED ON DEVICE 'D5'
*** NBR0740 COMMAND COMPLETED 'ATTACH-DEVICE';
***          (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
***
% VMS4005 DEVICE 'D4' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
          (BY GUEST) _____ (14)
% VMS4005 DEVICE 'X2' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2) (BY GUEST)
% VMS4005 DEVICE 'X3' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2) (BY GUEST)
% VMS4005 DEVICE 'D3' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2) (BY GUEST)
% VMS4005 DEVICE 'D5' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2) (BY GUEST)

```

- (12) The guest system operator uses the BS2000 command /ATTACH-DEVICE to attach the pubset. The devices of the pubset are assigned to VM2 implicitly here.
- (13) Various pubsets are detected in the guest system on the attached disks.
- (14) The VM and VM2000 administrators are notified about the implicit device assignment to VM2 by asynchronous messages.

```

/show-vm-device-status units=*std,information=*all,
    select=*par(assign-by-guest=*yes(set-name=sys2impl)) ----- (15)
UNIT DEV-TYPE PAV   S ASSIGN-SET  USAGE VM-ID/(#VM) ASSIGNED-BY
D3   D3435          SYS2IMPL  SH(I)   ( 2)   MIXED ----- (16)
                                1 MONITOR VM-ADMIN
                                2 SYSTEM2 GUEST
D4   D3435          SYS2IMPL(P) SH(D)   2 SYSTEM2 GUEST ----- (17)
D5   D3435          SYS2IMPL  SH(D)   2 SYSTEM2 GUEST
X2   HNC            SYS2IMPL  EXCL    2 SYSTEM2 GUEST ----- (18)
X3   HNC            SYS2IMPL  EXCL    2 SYSTEM2 GUEST
/show-vm-attributes information=*all,vm-id=system2 ----- (19)
CPU-POOL(P)
VM-ID      VM-GROUP(G) CPU-Q    MAX-U    PROC    MEM-SIZE  MAX-I/O
  2 SYSTEM2 *STDPOOL(P) 30.00   100.00  1(+1)   128 MB (100)

VM-ID      STATE          GUEST   VERSION ADMIN PRIV
  2 SYSTEM2 RUNNING        BS2-OSD V09.0A NO   AG,AS

VM-ID      CPU-POOL  GR-CPU-Q  CPU-Q/  EFF-Q  MAX-U/  EFF-U  PROC
  2 SYSTEM2 *STDPOOL  30.00/  25.00  100.00/ 25.00  1(+1)

VM-ID      SCHED ACT-I  VIRT CPUS
  2 SYSTEM2 TS    NO    00(RUN)

VM-ID      VM-MONJV
  2 SYSTEM2 NO VM-MONJV ASSIGNED

```

(Output continued on the next page)

- (15) The VM2000 administrator obtains detailed information on the status of the devices. If `INFORMATION=*ALL` is specified for a device which is assigned to multiple VMs for shared use, all of the VMs are output.
- (16) Disk D3 is assigned to the monitor VM and VM2. The usage mode is therefore `SH(I)` (SHARED with indirect I/O).
- (17) In the case of implicit device assignment, the disk devices are assigned in the usage mode `SHARED`. Input/output takes place by means of direct I/O because the devices are initially assigned to only one VM.
- (18) In the case of implicit device assignment, other devices (see [page 289](#)) are assigned in the usage mode `EXCLUSIVE`.
- (19) This command provides a VM administrator with information on his or her VM.


```

VM-ID      UNITS _____ (20)
  2 SYSTEM2  ASSIGNED BY VM-ADMIN
              EXCL:  D2,Z2,Z3
              ASSIGNED BY GUEST
              EXCL:  X2,X3
              SHARED: D3,D4,D5
...
VM-ID      PRIV  DETAILS
  2 SYSTEM2  AG    SYS2IMPL _____ (21)
              AS
***
*** Guest system operation on VM2:
*** /import-pubset pubset=vm04,use=*exclusive _____ (22)
*** DMS035B IMPORT PUBSET TASK WITH TSN 'XABT' FOR PUBSET WITH PUBSET ID
***          'VM04' HAS BEEN CREATED AND STARTED
*** SRM5017 USER GROUP MANAGEMENT ON PUBSET 'VM04' INITIALIZED WITH '1'
***          USER GROUP(S)
*** PRO6000 GUARDS CATALOG ':VM04:$TSOS.SYSCAT.GUARDS' OPENED
*** DMS038F IMPORT PUBSET PROCESSING COMPLETED . PUBSET WITH PUBSET ID
***          'VM04' IN 'LOCAL-IMPORTED' STATE
*** NBR0740 COMMAND COMPLETED 'IMPORT-PUBSET';
***          (RESULT: SC2=000, SC1=000, MC=DMS038F); DATE: <date>
*** /show-device-configuration unit=*pubset-devices(pubset=vm04),
***                               information=*vm2000 _____ (23)
*** MNEM UN-CLASS UN-TYPE CONF-STATE  VM-ASSIGN  VM-USAGE _____ (24)
*** D4  DVC      D3435  ATTACHED    BY-GUEST   EXCL _____ (25)
*** NBR0740 COMMAND COMPLETED 'SHOW-DEVICE-CONFIGURATION';
***          (RESULT:SC2=000, SC1=000, MC=CMD0001); DATE: <date>
***

```

- (20) The type of device assignment (BY VM-ADMIN, BY GUEST) and the usage mode (EXCL, SHARED) are displayed for VM2.
- (21) The ASSIGN-BY-GUEST privilege for the VM and the assignment set assigned are displayed.
- (22) In the guest system, pubset VM04 is imported in exclusive access mode. The pubset consists of disk device D4.
- (23) The guest system operator obtains information on the VM2000 status of the implicitly assigned disk devices of his or her guest system.
- (24) The output columns VM-ASSIGN and VM-USAGE also inform the guest system operator about the type of assignment and usage.
- (25) For the disk devices D4, the exclusive importing of the pubset has changed the usage mode from SHARED to EXCLUSIVE.

Changing the usage mode

```
/show-vm-device-status units=*std,information=*all,
      select=*par(state=*assigned(vm-id=system2)) _____ (26)
```

UNIT	DEV-TYPE	PAV	S	ASSIGN-SET	USAGE	VM-ID/(#VM)	ASSIGNED-BY
D2	D3435				EXCL	2 SYSTEM2	VM-ADMIN(P)
D3	D3435			SYS2IMPL	SH(I)	(2)	MIXED
						1 MONITOR	VM-ADMIN
						2 SYSTEM2	GUEST
D4	D3435			SYS2IMPL(P)	EXCL	2 SYSTEM2	GUEST _____ (27)
D5	D3435			SYS2IMPL	SH(D)	2 SYSTEM2	GUEST
Z2	SKP2				EXCL	2 SYSTEM2	VM-ADMIN
Z3	SKP2				EXCL	2 SYSTEM2	VM-ADMIN
X2	HNC			SYS2IMPL	EXCL	2 SYSTEM2	GUEST
X3	HNC			SYS2IMPL	EXCL	2 SYSTEM2	GUEST

```
/modify-vm-device-usage vm-id=system2,usage=*shared,
      units=*pubset-devices(pubset=*by-pubres-device(unit=d4)) _____ (28)
```

```
VMS4230 DEVICE USAGE OF PUBSET 'VM04' WITH PUBRES DEVICE 'D4' MODIFIED
```

```
/show-vm-device-status
      units=*pubset-devices(pubset=*by-pubres-device(unit=d4))
UNIT DEV-TYPE PAV S ASSIGN-SET USAGE VM-ID/(#VM) ASSIGNED-BY
D4 D3435 SYS2IMPL(P) SH(D) 2 SYSTEM2 GUEST _____ (29)
```

- (26) The VM2000 administrator obtains detailed information on the status of the devices assigned to VM2.
- (27) For the disks D4, the exclusive importing of the pubset has changed the usage mode from SHARED to EXCLUSIVE, see (25).
- (28) The VM2000 administrator nevertheless wants to set the usage mode SHARED for these devices again.
- (29) In spite of the exclusive use of the pubset, SHARED usage is set again in the guest system.

Implicit pubset release

```

***
*** Guest system operation on VM2:
*** /export-pubset pubset=vm04 _____ (30)
*** DMS035D EXPORT-PUBSET TASK WITH TSN 'XABV' FOR PUBSET WITH PUBSET ID
*** 'VM04' HAS BEEN CREATED AND STARTED
*** PRO6005 GUARDS CATALOG ':VM04:$TSOS.SYSCAT.GUARDS' CLOSED
*** DMS037F EXPORT-PUBSET PROCESSING COMPLETED FOR PUBSET 'VM04'
*** NBR0740 COMMAND COMPLETED 'EXPORT-PUBSET';
*** (RESULT: SC2=000, SC1=000, MC=DMS037F); DATE: <date>
*** /detach-device unit=*pubset-devices(pubset=vm04) _____ (31)
*** NKR0092 'DEVICE =D4': /DETACH-DEVICE PENDING ACCEPTED
*** NKR0045 *** 'DEVICE =D4' DETACHED ***
*** NBR0740 COMMAND COMPLETED 'DETACH-DEVICE';
*** (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
***
% VMS4009 DEVICE 'D4' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
(BY GUEST) _____ (32)
/show-vm-device-status
      units=*pubset-devices(pubset=*by-pubres-device(unit=e057)),
      information=*all
UNIT DEV-TYPE PAV S ASSIGN-SET USAGE VM-ID/(#VM) ASSIGNED-BY _____ (33)
D4 D3435 SYS2IMPL(P)

```

- (30) Pubset VM04 is exported in the guest system.
- (31) The pubset device(s) are detached in the guest system and consequently removed from the VM.
- (32) VM and VM2000 administrators are informed of the implicit device release by asynchronous messages.
- (33) The device has its original status again.

Converting from explicit to implicit pubset assignment

```
/add-vm-devices units=*pubset-devices(pubset=*by-pubres-device(unit=d4)),  
vm-id=system2 _____ (34)
```

```
VMS4005 DEVICE 'D4' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)  
VMS4220 PUBSET 'VM04' WITH PUBRES DEVICE 'D4' ASSIGNED TO VM (2,SYSTEM2)
```

```
/show-vm-device-status information=*std,  
units=*pubset-devices(pubset=*by-pubres-device(unit=d4))
```

```
UNIT DEV-TYPE PAV S ASSIGN-SET USAGE VM-ID/(#VM) ASSIGNED-BY _____ (35)  
D4 D3435 SYS2IMPL(P) EXCL 2 SYSTEM2 VM-ADMIN(P)  
***
```

```
*** Guest system operation on VM2:
```

```
*** /attach-device  
unit=*pubset-devices(pubset=*by-pubres-device(unit=d4)) _____ (36)
```

```
*** NKRO042 'DEVICE =D4': ATTACH ACCEPTED  
*** NKRO040 'DEVICE =D4' ATTACHED  
*** NKVD010 DISK 'VM04.0' MOUNTED ON DEVICE 'D4'  
*** NBR0740 COMMAND COMPLETED 'ATTACH-DEVICE';  
*** (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>  
***
```

```
/remove-vm-devices vm-id=system2,  
units=*pubset-devices(pubset=*by-pubres-device(unit=d4)) _____ (37)
```

```
VMS4085 DEVICE 'D4' STILL ATTACHED IN GUEST SYSTEM. DEVICE NOT REMOVED  
VMS4223 VM2000 COMMAND /REMOVE-VM-DEVICES FOR PUBSET SPECIFIED BY  
PUBRES DEVICE OR CATID 'D4' REJECTED
```

```
/remove-vm-devices units=*pubset-devices(  
pubset=*by-pubres-device(unit=d4)),vm-id=system2,  
force=*no(wait-for-unassign=*yes) _____ (38)
```

```
VMS4087 DEVICE 'D4' REMAINS ASSIGNED TO VM (2,SYSTEM2) UNTIL DETACH  
IN GUEST SYSTEM HAS BEEN DONE  
VMS4235 SOME DEVICES OF PUBSET 'VM04' WITH PUBRES DEVICE 'D4' REMAIN  
ASSIGNED TO VM (2,SYSTEM2) UNTIL DETACH IN GUEST SYSTEM HAS BEEN DONE
```

- (34) The VM2000 administrator assigns the device(s) of the pubset explicitly and exclusively to VM2.
- (35) The output of /SHOW-VM-DEVICE-STATUS indicates this. For the device D4 the PUBRES-FOR-ASSIGNMENT attribute is displayed in the ASSIGNED-BY column.
- (36) The devices are attached in the guest system.
- (37) The VM2000 administrator cannot remove the devices because they are still attached in the guest system.
- (38) By means of the FORCE=*NO(WAIT-FOR-UNASSIGN=*YES) operand the VM2000 administrator changes explicit assignment to implicit assignment (FORCE=*YES should only be used in exceptional cases).

```

/show-vm-device-status information=*std,
    units=*pubset-devices(pubset=*by-pubres-device(unit=d4))
UNIT DEV-TYPE PAV   S ASSIGN-SET  USAGE VM-ID/(#VM) ASSIGNED-BY _____ (39)
D4   D3435          SYS2IMPL(P) SH(D)  2 SYSTEM2  GUEST
***
*** Guest system operation on VM2:
*** /detach-device unit=*pubset-devices(pubset=vm04) _____ (40)
*** NKRO092 'DEVICE      =D4': /DETACH-DEVICE PENDING ACCEPTED
*** NKRO045 *** 'DEVICE      =D4' DETACHED ***
*** NBR0740 COMMAND COMPLETED 'DETACH-DEVICE';
***          (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
***
%   VMS4009 DEVICE 'D4' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
      (BY GUEST) _____ (41)
/show-vm-device-status units=*pubset-devices(
    pubset=*by-pubres-device(unit=d4)), information=*std
UNIT DEV-TYPE PAV   S ASSIGN-SET  USAGE VM-ID/(#VM) ASSIGNED-BY _____ (42)
D4   D3435          SYS2IMPL(P)

```

- (39) The output of /SHOW-VM-DEVICE-STATUS indicates that the device is now assigned implicitly to the VM. The PUBRES-FOR-ASSIGNMENT attribute (ASSIGNED-BY column) has once more been removed from the pubres device for this VM. At the same time the usage mode has also changed (SH(D)) because the pubset in the guest system has not been imported exclusively.
- (40) The device(s) are detached in the guest system and thus implicitly released.
- (41) VM and VM2000 administrators are informed of the implicit device release by asynchronous messages.
- (42) The device is free again.

Implicit device release at shutdown

```

/shutdown-vm vm-id=system2,action=*shutdown(quiet-time=0) _____ (43)
VMS2054 SHUTDOWN OF GUEST SYSTEM ON VM (2,SYSTEM2) INITIATED
% VMS2051 GUEST SYSTEM ON VM (2,SYSTEM2) DOWN; REASON: SHUTDOWN
% VMS4009 DEVICE 'D3' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2) _____ (44)
% VMS4009 DEVICE 'D5' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
% VMS4009 DEVICE 'X2' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
% VMS4009 DEVICE 'X3' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
/show-vm-attributes vm-id=system2,information=*all
CPU-POOL(P)
VM-ID      VM-GROUP(G)  CPU-Q    MAX-U    PROC    MEM-SIZE  MAX-IO
  2 SYSTEM2  *STDPOOL(P)  30.00   100.00   1(+1)    128 MB   (100)

VM-ID      STATE          GUEST   VERSION  ADMIN  PRIV
  2 SYSTEM2  DOWN          BS2-OSD V09.0A  NO     AG,AS _____ (45)

VM-ID      CPU-POOL  GR-CPU-Q  CPU-Q/  EFF-Q  MAX-U/  EFF-U  PROC
  2 SYSTEM2  *STDPOOL          30.00/  25.00  100.00/ 25.00  1(+1)

VM-ID      SCHED  ACT-I  VIRT  CPUS
  2 SYSTEM2  TS     NO    00(IDLE)

VM-ID      VM-MONJV
  2 SYSTEM2  NO VM-MONJV ASSIGNED

VM-ID      UNITS
  2 SYSTEM2  ASSIGNED BY VM-ADMIN
                EXCL:  D2,Z2,Z3 _____ (46)
...

```

- (43) The VM2000 administrator initiates the shutdown in the guest system.
- (44) The remaining, implicitly assigned devices of VM2 are released (unsolicited messages to the VM and VM2000 administrators).
- (45) The concluding information command shows that the guest system on VM2 is no longer active.
- (46) Only the explicitly assigned devices are still with VM2.

Explicit device release upon termination of the VM

```

/delete-vm vm-id=system2 _____ (47)
VMS4009 DEVICE 'Z2' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4009 DEVICE 'Z3' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4009 DEVICE 'D2' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4619 VIRTUAL MACHINE (2,SYSTEM2) REMOVED FROM CPU POOL '*STDPOOL'
VMS4024 VIRTUAL MACHINE (2,SYSTEM2) TERMINATED
/show-vm-assignment-set set-name=sys2impl _____ (48)
ASSIGNMENT-SET #DEVICES VM-ID
SYS2IMPL 5 NONE
/modify-vm-device-attributes units=*by-assignment-set(set-name=sys2impl),
assign-by-guest=*no _____ (49)
/show-vm-assignment-set set-name=sys2impl _____ (50)
ASSIGNMENT-SET #DEVICES VM-ID
SYS2IMPL 0 NONE
/delete-vm-assignment-set set-name=sys2impl _____ (51)
VMS4205 ASSIGNMENT SET 'SYS2IMPL' DELETED
/show-vm-assignment-set _____ (52)
ASSIGNMENT-SET #DEVICES VM-ID
*STDSET 0 NONE

```

- (47) VM2 is terminated. The explicitly assigned devices are removed.
- (48) Assignment set SYS2IMPL is no longer assigned to a VM, but it still contains 5 devices.
- (49) Implicit device assignment is no longer to be permitted for the devices in assignment set SYS2IMPL.
- (50) Assignment set SYS2IMPL now no longer contains devices; it can be deleted.
- (51) Assignment set SYS2IMPL is deleted.
- (52) Only the assignment set *STDSET still exists.

6.4.3 Special devices

Virtual console

It is advisable to operate guest systems via KVP consoles or logical consoles, see [page 100](#).

The operating strategy of VM2000 on S servers also provides for an emulation of the guest system console by the VM2000 hypervisor (see [section “Operating the guest systems\(VC dialog via \\$VMCONS\)” on page 117](#)).



Virtual consoles must be assigned to the VM explicitly (`TYPE=*VC`). The attribute `ASSIGN-BY-GUEST=*YES` cannot be set for virtual consoles.



On SQ servers virtual consoles (as generated devices) are not available. The VC dialog via `$VMCONS` is nevertheless possible.

Dummy tape devices

Dummy tape devices defined by systems support are used as backup devices in tape device management in BS2000/OSD. The attribute `ASSIGN-BY-GUEST=*YES` should not be set for them for implicit device assignment. The device can thus be attached in several guest systems simultaneously (`/ATTACH-DEVICE`) without being assigned to the VM. No inputs/outputs are executed by users to dummy tape devices.

Communication devices

To connect a guest system to a local area network (LOCLAN / ZASLAN / BRGLAN) and/or to a wide area network (WAN), the same options are available as for native operation.

For details on how to configure the communication devices, see [section “Local and remote communication” on page 64](#).

Snap units (virtual devices)

Snap units are virtual devices of the disk storage systems. They create a “Copy on Write” copy of a disk. A copy of this type enables the original disk to be restored.

Snap units are configured/generated in the disk storage system and when the server hardware is generated (using IOGEN (S servers) or X2000 (SQ servers)).

VM2000 detects snap units in Symmetrix and ETERNUS DX disk storage systems.

All snap units detected by VM2000 are marked with an “S” in the S column of the information output for `/SHOW-VM-DEVICE-STATUS`. This command can also be used to search specifically for snap units.



Snap units in other disk storage systems are not always recognized by VM2000 and flagged in the information output.

Guest systems (as of BS2000/OSD V7.0) can use snap units in two variants:

- Using Snapsets

A Snapset is a backup copy of a pubset which was created on snap units in the disk storage system. Each pubset disk corresponds to a snap unit. Snapsets are managed using BS2000 commands (e.g. `/RESTORE-FROM-SNAPSET`). Snap units are automatically selected for creating Snapsets by BS2000/OSD. Snap units are attached automatically when the pubset is imported and, under VM2000, implicitly assigned to the VM.

Under VM2000 the `AUTO-SNAP-ASSIGNMENT` privilege enables the guest system to implicitly assign itself snap units of a Snapset on a VM without the VM and device having to be prepared for the implicit device assignment (i.e. no `ASSIGN-BY-GUEST` privilege or attribute for the VM and device).

By default, a VM is assigned the `AUTO-SNAP-ASSIGNMENT` privilege with `/CREATE-VM`. The privilege is displayed with `AS` in `/SHOW-VM-ATTRIBUTES/-RESOURCES INFORMATION=*PRIVILEGES`.

- Snap session via SHC-OSD

In snap sessions under SHC-OSD, snap pairs (original unit and snap unit) are created by assigning one snap unit to each device which is to be mirrored. A corresponding number of snap units can also be assigned to a pubset (or to the pubset devices). Snap units can also be selected automatically (`*AUTO-SELECT` operand value).

Under VM2000 the snap units of a snap session via SHC-OSD must always satisfy the conditions for implicit device assignment or be assigned explicitly to the VM.

6.4.4 Shared disk devices

VM2000 supports both SPVS mode (SPVS=shared pubset) and SPD mode (SPD=shareable private disk) of disks.

When HIPLEX MSCF and a corresponding hardware configuration are used, simultaneous access to a shareable pubset is possible via several BS2000/OSD servers (see the “HIPLEX MSCF” manual [9]).

Depending on whether the virtual machines wishing to share these disks are set up on the same server or are distributed across several servers in a multiprocessor network, the VM2000 administrator must take different measures which are described on the pages below.

In a multiprocessor network, in addition to servers with VM2000 operation, servers in native mode which want to work with shareable disks can also be operated in parallel.

At any one time up to 16 guest systems or servers can work with a shared pubset and up to 4 guest systems or servers can work with a SPD disk.

Input/output handling for shared disks

See [page 293](#).

Sharing of disks by several VMs on one server

The shared use of a disk device by different VMs on a server is initiated by implicit or explicit device assignment to the VM in `SHARED` usage mode, see [page 292](#) (e.g. with `/ADD-VM-DEVICES . . . ,TYPE=*SD`). The `SHARED` usage mode is retained at `/SWITCH-VM-DEVICES`.

Desired configuration:

VMx and VMy wish to use the disks `DISK01`, `DISK02` and `DISK03` for `SPVS` or `SPD` mode.

Action to be taken by the VM2000 administrator:

The disk devices from `DISK01` to `DISK03` must be assigned to each VM as shareable, e.g. with the commands:

```
/ADD-VM-DEVICES UNITS=(DISK01,DISK02,DISK03),VM-ID=VMx,TYPE=*SD
/ADD-VM-DEVICES UNITS=(DISK01,DISK02,DISK03),VM-ID=VMy,TYPE=*SD
```

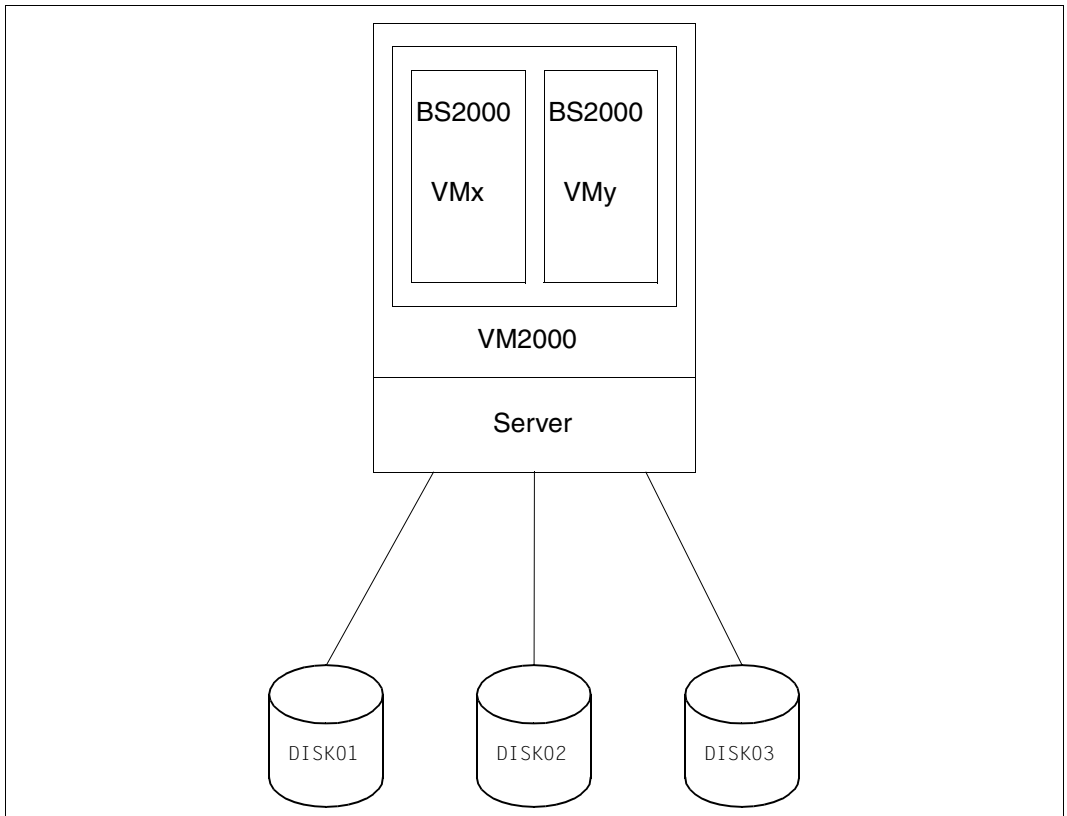


Figure 23: Sharing of disks by several VMs on one server

Sharing of disks in a VM2000 multiprocessor network

In a multiprocessor network, several servers (possibly from different server series) with VM2000 are operated in parallel. Each server can access shared peripheral devices. The BS2000 systems run as guest systems under VM2000 on the servers.



Disk devices of channel peripherals can be shared by S servers.
 Disk devices of FC peripherals can be shared by SQ servers.
 Disks of type D3435 can be shared by S and SQ servers if they are connected to the S server via a type FC channel.

The following configurations may be possible:

a) Desired configuration:

On server 1 only VMm and on server 2 only VMy wish to use the disks DISK01, DISK02 and DISK03 for SPVS or SPD mode (see [figure 24 on page 341](#)).

Action to be taken by the VM2000 administrator:

The disk devices from DISK01 to DISK03 should be assigned to VMm and VMy **exclusively** with the command:

```
/ADD-VM-DEVICES UNITS=(DISK01,DISK02,DISK03),VM-ID=VMm,TYPE=*NONE
/ADD-VM-DEVICES UNITS=(DISK01,DISK02,DISK03),VM-ID=VMy,TYPE=*NONE
```

b) Desired configuration:

On server 1 only VMm and on server 2 both VMs wish to use the disks DISK01, DISK02 and DISK03 for SPVS or SPD mode (see [figure 24 on page 341](#)).

Action to be taken by the VM2000 administrator:

- The disk devices from DISK01 to DISK03 should be assigned to VMm **exclusively** with the command:

```
/ADD-VM-DEVICES UNITS=(DISK01,DISK02,DISK03),VM-ID=VMm,TYPE=*NONE
```

- The disk devices from DISK01 to DISK03 must be assigned to both VMs (VMx and VMy) as **shareable**, e.g. with the command:

```
/ADD-VM-DEVICES UNITS=(DISK01,DISK02,DISK03),VM-ID=VMx,TYPE=*SD
/ADD-VM-DEVICES UNITS=(DISK01,DISK02,DISK03),VM-ID=VMy,TYPE=*SD
```

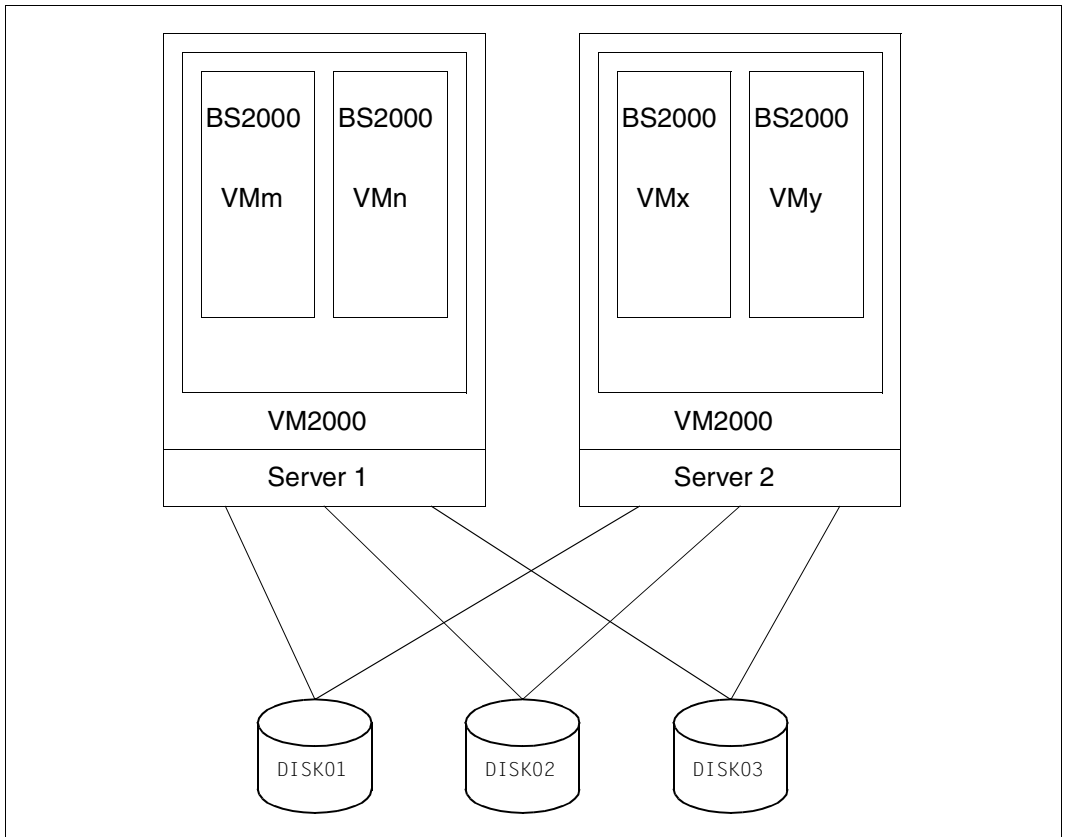


Figure 24: Sharing of disks by VMs on different servers

Sharing of disks in a mixed multiprocessor network

This section applies to configurations that meet the following **general** criteria:

- SPD or SPVS mode between several BS2000 systems on different servers of different server series
- at least one of the BS2000 systems active in SPD or SPVS mode is running as a guest system under VM2000
- at least one of the BS2000 systems active in SPD or SPVS mode is running in native mode



Disk devices of channel peripherals can be shared by S servers.

Disk devices of FC peripherals can be shared by SQ servers.

Disks of type D3435 can be shared by S and SQ servers if they are connected to the S server via a type FC channel.

For shared disk devices of channel peripherals (S servers), it should additionally be noted that:

- one or more of the disks with channel connection for SPD or SPVS operation are connected via the same channel director
- the “dynamic reconnection feature” of this channel director may be used.

The “dynamic reconnection feature” requires that the partners are uniquely identified on the HSI when several partners (servers) are effecting access. This means that each partner has a unique ID, known as the “path group ID”. All guest systems of a VM2000 system use the same “path group ID” which is unique on all the servers involved.



The example below also applies for disk devices in the channel peripherals when the "path group ID" and "dynamic reconnection feature" are taken into consideration.

Example of a mixed multiprocessor network

The [figure 25 on page 344](#) shows an example of a configuration of the FC peripherals in which in addition to a native-mode BS2000 system, two other server (on each of which several BS2000 systems are running as guest systems under VM2000) access shareable disk devices.

Desired configuration:

On server 1 only VMm and on server 2 both VMs wish to use the disks `DISK01`, `DISK02` and `DISK03` for SPVS or SPD mode. On server 3 a BS2000 system is operated without VM2000, and, together with the VMs on the other server, it wishes to use the disks for SPVS or SPD mode.

Action to be taken by the VM2000 administrator:

- The disk devices from `DISK01` to `DISK03` should be assigned to VMm on server 1 **exclusively** with the command:

```
/ADD-VM-DEVICES UNITS=(DISK01,DISK02,DISK03),VM-ID=VMm,TYPE=*NONE
```
- The disk devices from `DISK01` to `DISK03` must be assigned to both VMs (VMx and VMy on server 2) as **shareable**, e.g. with the command:

```
/ADD-VM-DEVICES UNITS=(DISK01,DISK02,DISK03),VM-ID=VMx,TYPE=*SD  
/ADD-VM-DEVICES UNITS=(DISK01,DISK02,DISK03),VM-ID=VMy,TYPE=*SD
```
- BS2000/OSD on server 3 acts like a BS2000/OSD in the native network (see also the “HIPLEX MSCF” manual [\[9\]](#)).

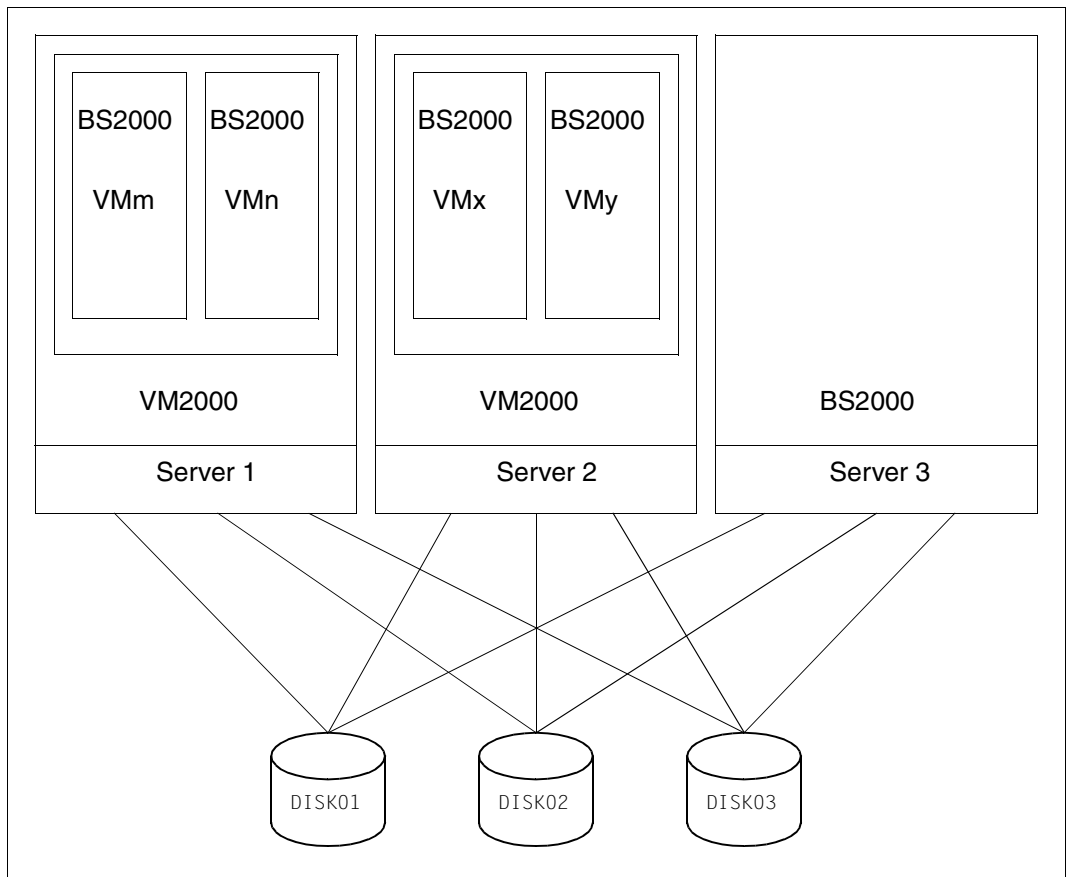


Figure 25: Sharing of channel disks in the mixed multiprocessor network

6.4.5 Parallel access volume (PAV, S server)

Input/output requests of different applications to the same device are generally serialized. Input/output requests that come later have to wait until the previous input/output is completed. Up to now it has only been possible to improve performance in this respect by means of appropriate data organization on different data carriers.

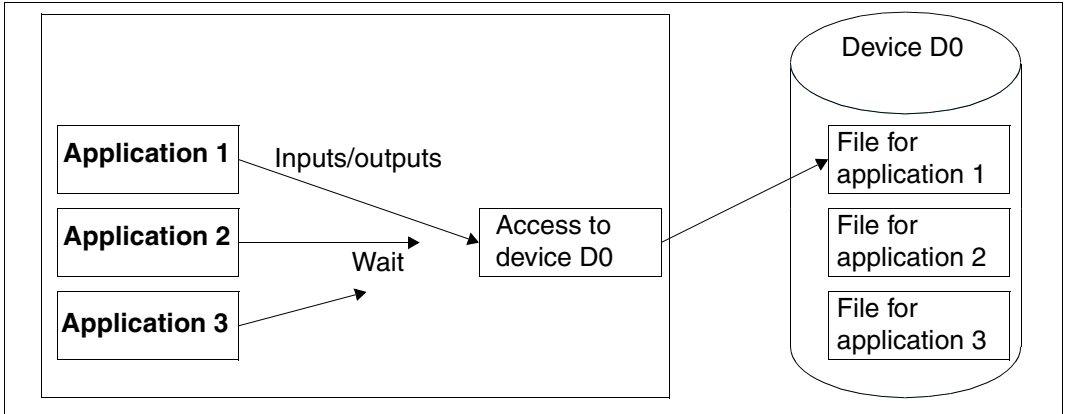


Figure 26: Applications with input/output load on one device (serialized)

For ETERNUS DX400/DX8000 disk storage systems and for Symmetrix systems of the model series 8xxx and DMX, BS2000/OSD offers the function Parallel Access Volume (PAV) which enables multiple I/Os to a disk device (a logical volume) to be performed in parallel.

SQ PAV devices are not available on SQ servers, where the inputs/outputs are automatically parallelized by X2000.

A PAV device consists of a real **base device** (BASE) with up to seven **alias devices** (ALIAS).

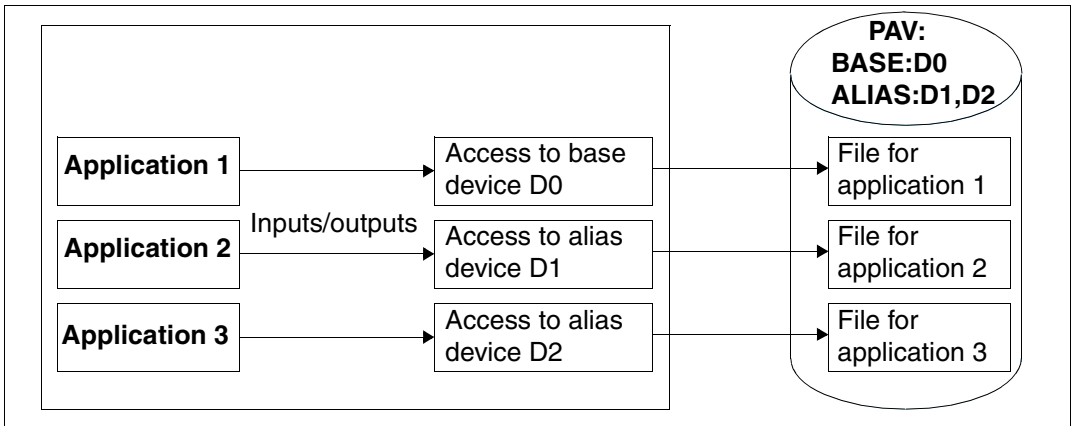


Figure 27: Applications with input/output load on one device (parallel access volume)

Further information on PAV is provided in the “Introductory Guide to Systems Support” [2].

(Static) PAV in native operation

The way PAVs are operated depends on the type of channel connection they have:

- Disk devices of the types D3435 in FBA format and D3490-xx in CKD format on the type S channel

Here the PAVs are set up in Symmetrix disk systems by the system service. The PAV property, i.e. the property which determines whether a device is a base device or an alias device, and the assignment of the alias devices to the base device is recognized by the operating system when a device is attached.

If changes are made to a PAV in the Symmetrix disk system by the system service staff, the base device must be detached beforehand. After the change has been made, the base device can be attached again. When it is attached, the (new) properties are detected by the operating system.

At hardware generation of channel peripherals (see the “System Installation” manual [1]), the base device and alias devices are generated as independent devices of the same type with different unit addresses and different mnemonic device names on the same controller.

- Disk devices of the type D3435 in FBA format on the type FC channel
PAV on the FC channel can be used for ETERNUS DX400/8000 and Symmetrix disk storage systems.

Here the PAVs are set up only when the IO configuration data is generated (DVC statement of IOGEN). The PAV properties are detected by the operating system from the device’s generation data. Changes to the PAV structure require generation to be performed again or dynamic modification of the IO configuration, see [page 348](#).

At hardware generation of the FC channel peripherals (see the “System Installation” manual [1]), the base device and alias devices are generated as independent devices of the same type with the same Logical Unit Numbers but different unit addresses and mnemonic device names on the same controller.

When the base device is attached, BS2000/OSD automatically attaches the associated alias devices. After the base and alias devices have been attached, I/Os for the PAV can be processed in parallel via base and alias devices. Device assignment in the guest system continues to apply only for the base device. I/Os are issued to the base device by the applications. The operating system (IOCNTRL component) controls execution via the base or alias device.

The BS2000 command `/SHOW-DEVICE-CONFIGURATION INFORMATION=*PAV` provides information on PAVs.

(Static) PAV under VM2000

PAVs can be used under VM2000 for the monitor system and the guest systems.

The base device can be assigned to a VM explicitly or implicitly (see [page 285](#)).

Alias devices are always assigned to a VM implicitly, regardless of the `ASSIGN-BY-GUEST` setting, if the base device is attached in the guest system. Explicit assignment of alias devices is rejected.



If a device (alias device) has been assigned to a VM explicitly because the “alias device” property cannot be detected, this assignment is reversed as soon as it is detected (message `VMS4192`).

Alias devices always have the same usage mode as the associated base device. The usage mode of a base device can be changed with `/MODIFY-VM-DEVICE-USAGE USAGE=*SHARED/*EXCLUSIVE` (the usage mode of its alias devices thus changes as well).

For information on I/Os for shared devices, see also [page 293](#).

An (implicitly or explicitly assigned) base device can be detached from a VM explicitly. In implicit release, the base device is released by the guest system (see [page 290](#)). Alias devices are released automatically with the base device. Explicit removal of alias devices is rejected.

When changes are made to a PAV in the Symmetrix disk system (disk devices on the type S channel), the base device must be detached (and removed from the VMs) in all the guest systems to which it is assigned. After the changes are made, it can be (assigned to the VMs and) attached again. At (assignment and) attachment, the (new) PAV properties are detected by VM2000 and the guest system.

You can obtain information on PAVs under VM2000 by means of `/SHOW-VM-DEVICE-STATUS`. The `PAV-ATTRIBUTE` operand lets you control the selection of PAVs. The `INFORMATION=*PAV` operand displays the PAV structure for the relevant device.

In order to obtain as even a distribution of I/Os as possible for shared disks under VM2000, the `IOCNTL` component in the guest systems prefers the base device or a special alias device. The selection is made via the VM index. Preferred base or alias devices can also be defined explicitly in each guest system by means of the BS2000 command (see [page 201](#))

```
/MODIFY-IO-UNIT UNIT=*DEVICE(NAME=...,STATE=*PAV-PREFERRED-DEVICE(...))
```

Dynamic PAV in native operation

Static PAV, as described above, requires careful planning of future device utilization. When generation in BS2000/OSD takes place, it must be borne in mind that a separate address is required for each alias device.

Dynamic PAV (the IORM function DPAV, see [page 359](#) and the “Utility Routines” manual [3]) requires fewer alias devices. As with static PAV, alias devices must be generated, but it is not necessary to provide the maximum number of alias devices required for each volume in advance. DPAV autonomously assigns alias devices to those volumes which profit from this the most.



The IORM function DPAV is offered for disk devices on type FC channels of the S servers in native mode and under VM2000.

Dynamic PAV under VM2000

DPAV under VM2000 requires the IORM subsystem in the monitor system.

When a bottleneck occurs for a PAV device in a guest system, the IORM subsystem in the guest system requests the IORM subsystem in the monitor system to coordinate and execute the following actions:

1. Search for the alias device which is utilized least.
2. Request all BS2000 guest systems to detach the selected alias device. Here the alias device is removed from the VM because an alias device is only implicitly assigned to a VM.
3. Assign the selected alias device to the new base device by means of a dynamic modification of the configuration (`/REMOVE-IO-UNIT`, `/ADD-IO-UNIT`).
4. Inform all BS2000 guest systems of the new assignment. In this case the newly configured alias device is also attached, and consequently implicitly assigned to its own VM, in the guest systems in which the base device is attached.

Example of the use of (static) PAV under VM2000

In this example a base device is assigned to the monitor system. The base device is attached and then detached again in the monitor system.

```

/show-vm-device-status units=*std,information=*pav,
                        select=*par(state=*disposable,pav-attribute=*base) ----- (1)
UNIT DEV-TYPE PAV   S ASSIGN-SET  USAGE VM-ID/(#VM) ASSIGNED-BY
2300 STDDISK  BASE ----- (2)
                        (BASE: 2300; ALIAS: 2310,2320,2330)
2301 STDDISK  BASE
                        (BASE: 2301; ALIAS: 2311,2321,2331)
2302 STDDISK  BASE
                        (BASE: 2302; ALIAS: 2312,2322,2332)
2303 STDDISK  BASE
                        (BASE: 2303; ALIAS: 2313,2323,2333)
2304 STDDISK  BASE
                        (BASE: 2304; ALIAS: 2314,2324,2334)
2305 STDDISK  BASE
                        (BASE: 2305; ALIAS: 2315,2325,2335)
... (Ausgabe gekürzt)
5400 STDDISK  BASE
                        (BASE: 5400; ALIAS: 540C)
/add-vm-devices units=2300,vm-id=monitor ----- (3)
VMS4005 DEVICE '2300' ASSIGNED TO VIRTUAL MACHINE (1,MONITOR)
/show-vm-device-status units=23*,information=*pav,
                        select=*par(state=*assigned(vm-id=monitor)) ----- (4)
UNIT DEV-TYPE PAV   S ASSIGN-SET  USAGE VM-ID/(#VM) ASSIGNED-BY
2300 STDDISK  BASE                EXCL  1 MONITOR  VM-ADMIN
                        (BASE: 2300; ALIAS: 2310,2320,2330)

```

- (1) The VM2000 queries which base devices are free. Detailed information is to be displayed on the configuration of the PAVs (base and alias devices).
- (2) The list of free base devices and their alias devices is output.
- (3) The base device 2300 is assigned explicitly to the monitor VM.
- (4) The VM2000 administrator displays all devices whose mnemonic names begin with 23 and are assigned to the monitor VM. Only the base device 2300 is currently assigned.

```

***
*** Guest system operation on the monitor VM:
*** /attach-device unit=2300 _____ (5)
*** NKRO042 'DEVICE      =2300': ATTACH ACCEPTED
*** NKRO042 'DEVICE      =2300': ATTACHED
*** NKRO042 'DEVICE      =2310': ATTACH ACCEPTED _____ (6)
*** NKRO042 'DEVICE      =2320': ATTACH ACCEPTED
*** NKRO042 'DEVICE      =2330': ATTACH ACCEPTED
*** NKVD010 DISK 'PF1.00' MOUNTED ON DEVICE '2300'
*** NBR0740 COMMAND COMPLETED 'ATTACH-DEVICE';
***          (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: 2006-11-13
*** NKRO073 ALIAS DEVICE '2310' LINKED TO BASE DEVICE '2300'
*** NKRO073 ALIAS DEVICE '2320' LINKED TO BASE DEVICE '2300'
*** NKRO073 ALIAS DEVICE '2330' LINKED TO BASE DEVICE '2300'
*** NKRO040 'DEVICE      =2310' ATTACHED
*** NKRO040 'DEVICE      =2320' ATTACHED
*** NKRO040 'DEVICE      =2330' ATTACHED
***
% VMS4005 DEVICE '2310' ASSIGNED TO VIRTUAL MACHINE (1,MONITOR)
          (BY GUEST) _____ (7)
% VMS4005 DEVICE '2320' ASSIGNED TO VIRTUAL MACHINE (1,MONITOR) (BY GUEST)
% VMS4005 DEVICE '2330' ASSIGNED TO VIRTUAL MACHINE (1,MONITOR) (BY GUEST)

```

```

/show-vm-device-status units=23*,information=*pav,
                        select=*par(state=*assigned(vm-id=monitor))
UNIT DEV-TYPE PAV   S ASSIGN-SET  USAGE VM-ID/(#VM) ASSIGNED-BY
2300 D3435     BASE           EXCL  1 MONITOR  VM-ADMIN
      (BASE: 2300; ALIAS: 2310,2320,2330)
2310 D3435     ALIAS           EXCL  1 MONITOR  GUEST _____ (8)
      (BASE: 2300; ALIAS: 2310,2320,2330)
2320 D3435     ALIAS           EXCL  1 MONITOR  GUEST
      (BASE: 2300; ALIAS: 2310,2320,2330)
2330 D3435     ALIAS           EXCL  1 MONITOR  GUEST
      (BASE: 2300; ALIAS: 2310,2320,2330)

```

- (5) The base device is attached in the monitor system.
- (6) The alias devices are thus assigned and attached to the monitor VM implicitly.
- (7) VM and VM2000 administrators are informed about the implicit device assignment by asynchronous messages.
- (8) The VM2000 administrator repeats the command from (4). Now the alias devices are assigned implicitly to the monitor VM with the same usage mode.

```

/show-vm-attributes vm-id=monitor,information=*units _____ (9)
VM-ID          UNITS
 1 MONITOR     ASSIGNED BY VM-ADMIN
                EXCL:   C2,C3,V0,V1,2300,2430,2431,2432,2433
                SHARED: 22F4,22F5,22F6,22F7,58F2
                ASSIGNED BY GUEST
                EXCL:   2310,2320,2330
***
*** Guest system operation on the monitor VM:
*** /detach-device unit=2300 _____ (10)
*** NKRO092 'DEVICE      =2300': /DETACH-DEVICE PENDING ACCEPTED
*** NKRO045 *** 'DEVICE      =2300' DETACHED ***
*** NBRO740 COMMAND COMPLETED 'DETACH-DEVICE';
***           (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: 2006-11-13
*** NKRO074 ALIAS DEVICE '2310' UNLINKED FROM BASE DEVICE '2300' _____ (11)
*** NKRO074 ALIAS DEVICE '2320' UNLINKED FROM BASE DEVICE '2300'
*** NKRO074 ALIAS DEVICE '2330' UNLINKED FROM BASE DEVICE '2300'
*** NKRO045 *** 'DEVICE      =2310' DETACHED ***
*** NKRO045 *** 'DEVICE      =2320' DETACHED ***
*** NKRO045 *** 'DEVICE      =2330' DETACHED ***
***
% VMS4009 DEVICE '2310' REMOVED FROM VIRTUAL MACHINE (1,MONITOR)
                (BY GUEST) _____ (12)
% VMS4009 DEVICE '2320' REMOVED FROM VIRTUAL MACHINE (1,MONITOR) (BY GUEST)
% VMS4009 DEVICE '2330' REMOVED FROM VIRTUAL MACHINE (1,MONITOR) (BY GUEST)

/show-vm-device-status units=23*,information=*pav,
                select=*par(state=*assigned(vm-id=monitor)) _____ (13)
UNIT DEV-TYPE PAV   S ASSIGN-SET  USAGE VM-ID/(#VM) ASSIGNED-BY
2300 D3435   BASE                EXCL 1 MONITOR VM-ADMIN
                (BASE: 2300; ALIAS: 2310,2320,2330)
/remove-vm-devices units=2300,vm-id=monitor
VMS4005 DEVICE '2300' REMOVED FROM VIRTUAL MACHINE (1,MONITOR)

```

- (9) /SHOW-VM-ATTRIBUTES provides a VM administrator with information on the devices of his or her VM.
- (10) The base device is detached again in the monitor system.
- (11) The alias devices are thus implicitly detached and released.
- (12) VM and VM2000 administrators are informed of the implicit device release by asynchronous messages.
- (13) Only the base device is assigned explicitly to the monitor VM.

6.4.6 Using the Highspeed Net Connect HNC (S server)

The Highspeed Network Connect HNC connects S servers with the LAN. The HNC offers a web interface, the HNC manager, to operate it.

Details of the HNC (HNC-IV 91853 and HNC-V 91854) are provided in the "HNC" manual [21].

With **one** HNC, network connections to the monitor system and all other guest systems can be established under VM2000.

The data transfer between the various BS2000 guest systems can take place internally in the HNC without placing a load on the network. This internal network can be used to enable a guest system to communicate with another guest system on the same server or with a guest system on another server (with two channel boards), without having to be routed via an external network.

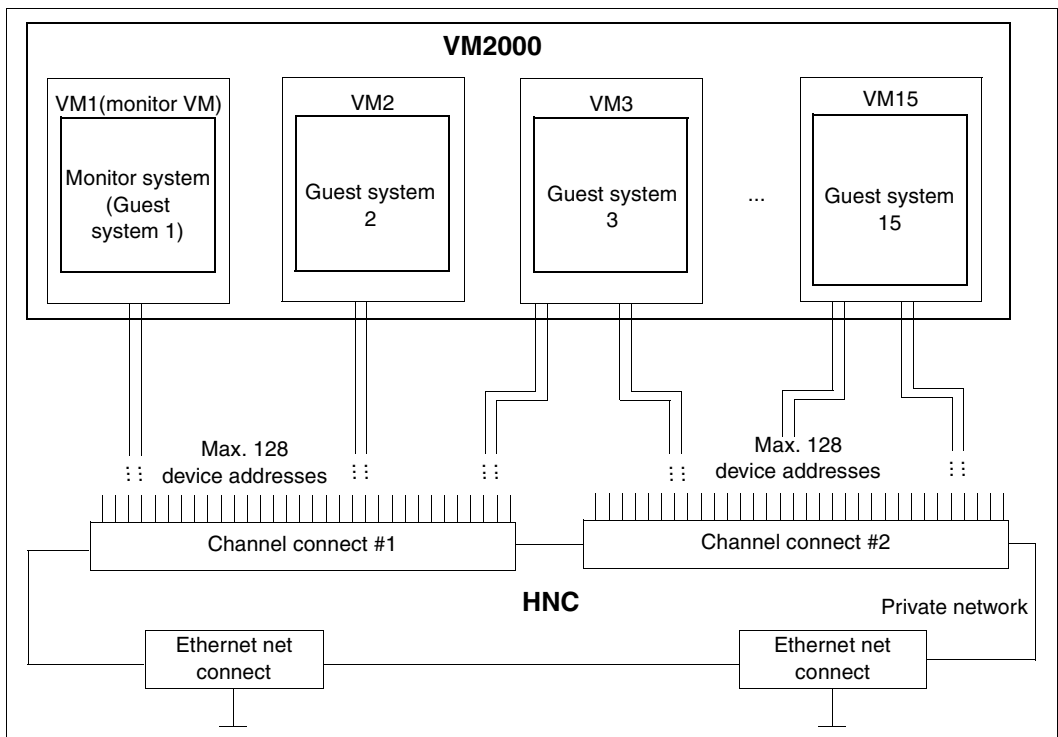


Figure 28: Using the HNC under VM2000 for 15 guest systems

If a special guest system is to be accessed by the HNC networks, a BCAM line adapter must be defined in this guest system for each network and a device address pair set up for it in the HNC.

This means that if, for example, three Ethernet networks are connected and each of these networks and the internal network is to be able to access all the configured guest systems, up to 32 guest systems can be accessed by these networks via an HNC:

- For each network there is one BCAM line adapter per guest system.
- With four different networks there are thus four BCAM line adapters per guest system.
- With a maximum of 128 possible BCAM line adapters, up to 32 guest systems can therefore be accessed.

In practice the number of guest systems used productively is lower. This means that a full network configuration can be operated with one HNC.

Specimen configuration

In the following example, such a configuration is assumed. The statements needed for installation, generation and assignment are shown (see [figure 29 on page 354](#)).

Three Ethernet networks and the internal network are installed. Six guest systems are running under VM2000:

- the monitor system (MONIT in VM1)
- two production systems (PROD1 in VM2, PROD2 in VM3)
- one maintenance system (MAINT1 in VM4)
- two test systems (TEST1 in VM5, TEST2 in VM6).

An HNC is installed on two channels (channel connect #1 (C0) and #2 (C1)).

32 device addresses (E0, E1, ... , E7, F0, F1, ... , FB, P0, P1, ..., PB), 16 per channel, are occupied. The assignment of the device address pairs to VMs, channel connects and networks can be seen from [figure 29 on page 354](#).

Example

The device address pair F0,F1 on channel connect #1 (C0) belongs to a monitor VM (monitor system) BCAM line adapter: F0,F1,(VM1,#1).

The production systems PROD1 and PROD2 are to be accessed from within three networks. The monitor system and the other guest systems are each to be accessible from within two networks.

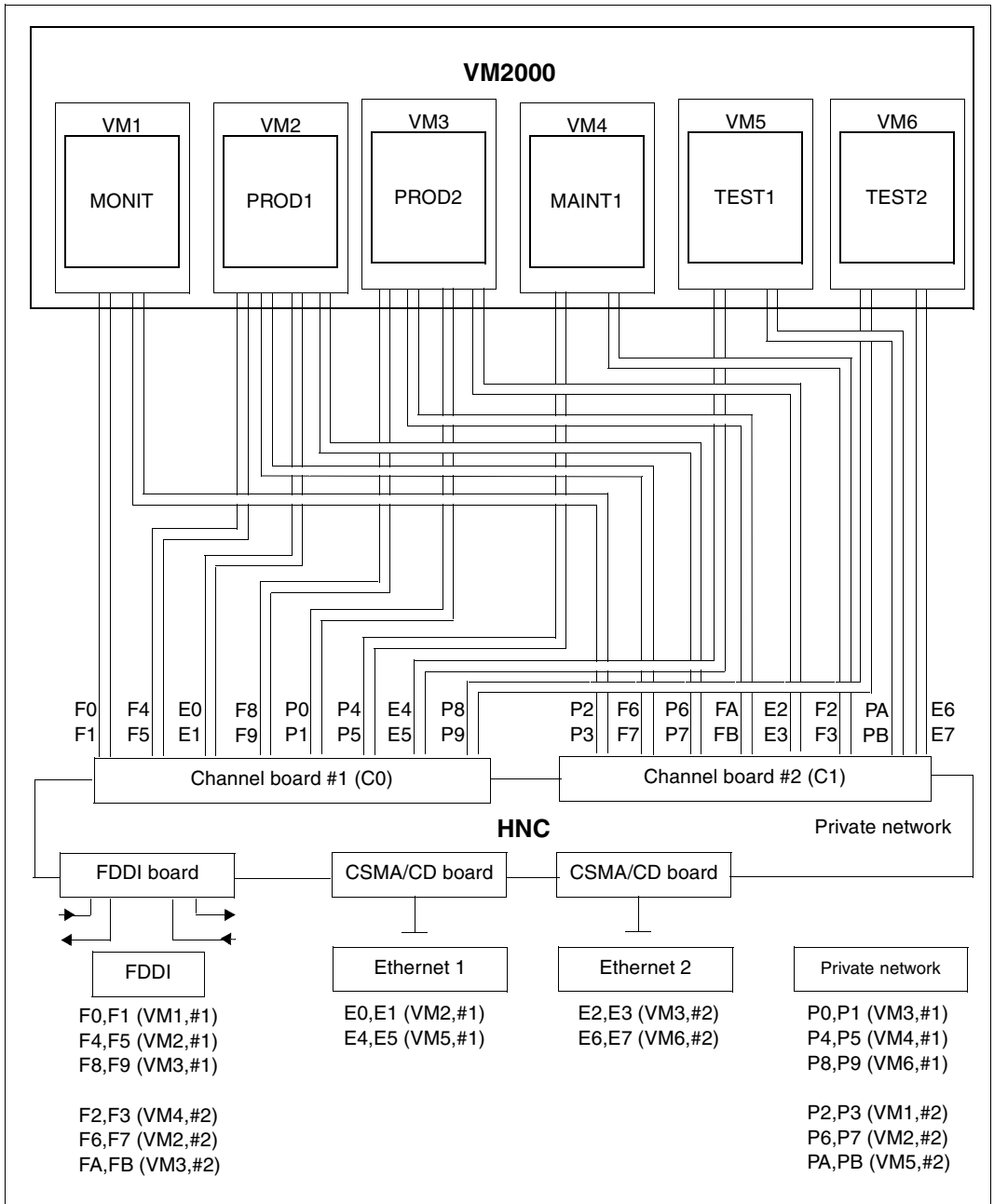


Figure 29: Specimen configuration for HNC

Installation of the specimen configuration

The device address pairs for the BCAM line adapters (assigning the network and a network address in each case) are set up by the service when the HNC is installed.

Generation of the specimen configuration

To generate the monitor system and the guest systems the appropriate generation statements must be specified for the channel trunks and for the total of 32 device addresses. In the above example these must be specified as follows when they are connected to, for example, channel K0 and K1:

- * Channel connect #1 of HNC
CTL C0,,(K0,0),TMR=S4
- * Channel connect #2 of HNC
CTL C1,,(K1,0),TMR=S4
- * ETHERNET via channel connect #1 (C0)
 - DVC F0,6D,A,00,(C0)
 - DVC F1,6D,A,01,(C0)
 - DVC F4,6D,A,02,(C0)
 - DVC F5,6D,A,03,(C0)
 - DVC F8,6D,A,04,(C0)
 - DVC F9,6D,A,05,(C0)
- * ETHERNET via channel connect #1 (C0)
 - DVC E0,6D,A,06,(C0)
 - DVC E1,6D,A,07,(C0)
 - DVC E4,6D,A,08,(C0)
 - DVC E5,6D,A,09,(C0)
- * Internal network via channel connect #1 (C0)
 - DVC P0,6D,A,0A,(C0)
 - DVC P1,6D,A,0B,(C0)
 - DVC P4,6D,A,0C,(C0)
 - DVC P5,6D,A,0D,(C0)
 - DVC P8,6D,A,0E,(C0)
 - DVC P9,6D,A,0F,(C0)
- * ETHERNET via channel connect #2 (C1)
 - DVC F2,6D,A,00,(C1)
 - DVC F3,6D,A,01,(C1)
 - DVC F6,6D,A,02,(C1)
 - DVC F7,6D,A,03,(C1)
 - DVC FA,6D,A,04,(C1)
 - DVC FB,6D,A,05,(C1)

- * ETHERNET via channel connect #2 (C1)
 - DVC E2,6D,A,06,(C1)
 - DVC E3,6D,A,07,(C1)
 - DVC E6,6D,A,08,(C1)
 - DVC E7,6D,A,09,(C1)

- * Private network via channel connect #2 (C1)
 - DVC P2,6D,A,0A,(C1)
 - DVC P3,6D,A,0B,(C1)
 - DVC P6,6D,A,0C,(C1)
 - DVC P7,6D,A,0D,(C1)
 - DVC PA,6D,A,0E,(C1)
 - DVC PB,6D,A,0F,(C1)

The BCAM line adapters to the device address pairs must be defined by generation (RDF files) or via the appropriate BCAM statements (in SOF files, for example) (see the “Introductory Guide to Systems Support” [2]).

Assigning the specimen configuration to VM2000

1. The monitor VM is set up and started as far as VM2000 READY. The information for setting up the monitor VM is stored in the configuration file of the monitor VM. While it is not essential, the network peripherals of the monitor should also be assigned via the configuration file, as they are then automatically available and “attached” when BCAM is started. The following statement is required for this in the above example:

```
/ADD-VM-DEVICES UNITS=(F0,F1,P2,P3),VM-ID=MONIT
```

2. Once VM2000 READY has been reached in the monitor system, it is possible to start installing the guest systems. The following shows the requisite statements for the network configuration:

```
/CREATE-VM VM-INDEX=2,VM-NAME=PROD1,MEMORY-SIZE=....  
/ADD-VM-DEVICES UNITS=(F4,F5,F6,F7,E0,E1,P6,P7),VM-ID=PROD1  
...  
  
/CREATE-VM VM-INDEX=3,VM-NAME=PROD2,MEMORY-SIZE=....  
/ADD-VM-DEVICES UNITS=(F8,F9,FA,FB,E2,E3,P0,P1),VM-ID=PROD2  
...  
  
/CREATE-VM VM-INDEX=4,VM-NAME=MAINT1,MEMORY-SIZE=....  
/ADD-VM-DEVICES UNITS=(F2,F3,P4,P5),VM-ID=MAINT1  
...  
  
/CREATE-VM VM-INDEX=5,VM-NAME=TEST1,MEMORY-SIZE=....  
/ADD-VM-DEVICES UNITS=(E4,E5,PA,PB),VM-ID=TEST1  
...  
  
/CREATE-VM VM-INDEX=6,VM-NAME=TEST2,MEMORY-SIZE=....  
/ADD-VM-DEVICES UNITS=(E6,E7,P8,P9),VM-ID=TEST2  
...  
  
/START-VM VM-ID=PROD1,IPL-UNIT=..., ...  
/START-VM VM-ID=PROD2,IPL-UNIT=..., ...  
/START-VM VM-ID=MAINT1,IPL-UNIT=..., ...  
/START-VM VM-ID=TEST1,IPL-UNIT=..., ...  
/START-VM VM-ID=TEST2,IPL-UNIT=..., ...
```

6.4.7 Notes on configuration for VM2000 operation

In theory the peripheral devices belonging to a BS2000/OSD server can be freely distributed over the virtual machines. The following guidelines should be borne in mind when configuring the virtual machines:

Notes on configuration for all server series

- The operation of the guest system on a VM (operating) is normally carried out by means of BS2000 consoles (see [page 100](#)).

Alternatively guest systems can also be operated via \$VMCONS.

- In each guest system (and the monitor system as well) the configuration statuses of the KVP devices that are not used as a KVP console in the relevant guest system should be set to DETACHED via the startup parameter service (IOCONF parameter set, MODIFY-IO-UNIT statement). The KVP devices used in the guest system should remain ATTACHED (default).

Notes on configuration for S servers

- If different virtual machines are working with channel devices that are attached to **one** device controller, heavy I/O utilization of these devices may result in bottlenecks. To avoid this, all the devices attached to the same device controller should be assigned to **one** virtual machine wherever possible.
- Under VM2000, disk devices can be assigned to the VMs for shared use. If a disk device is shared by at least two VMs on an S server, all of the input/output traffic of the shared disk devices is coordinated by the VM2000 hypervisor. On S servers disk devices should therefore not be operated as shareable unnecessarily.

6.4.8 Use of IORM in VM2000 operation

The BS2000 utility routine IORM (IO Resource Manager) offers functions for autonomous, dynamic control of the input/output resources channel, controller, path and device:

- IOPT: I/O Priority handling for Tasks
- DPAV: Dynamic Parallel Access Volume
- DDAL: Optimized Dynamic Device Allocation in CentricStor mode
- TCOM: Dynamic Tape Compression
- IOLVM: I/O Limit for Virtual Machines

SQ The IORM functions DPAV and IOLVM have no relevance for SQ servers.

During ongoing operation, IORM collects data on the utilization of the I/O resources and controls I/O operation in accordance with the specified threshold values.

You will find a detailed description of IORM in the “Utility Routines” manual [3].

IORM works on a cross-VM basis but not on a cross-server basis.

In VM2000 operation, IORM should be started on the monitor system and on all BS2000 guest systems. IORM is administered via the monitor system. The central data storage of IORM is also located there. The IORM subsystems on the monitor system and on the BS2000 guest systems communicate via an internal interface.

In VM2000 operation, the functions of IORM are used for selected VMs or they apply globally for all VMs on a server:

- IOPT
The IOPT function is enabled on the guest systems on which IOPT is to be active.
- DPAV (S servers)
The DPAV function must be enabled on the monitor system and on the guest systems on which DPAV is to be active. The actual switchover of alias devices is coordinated and executed by DPAV on the monitor system, see [page 348](#).
- DDAL
This function extends the optimized (local) device selection in native mode to all guest systems of a server in VM2000 operation. For this purpose it is important that IORM is started on the monitor system and on all BS2000 guest systems and that the DDAL function is enabled on all of these.
The DDAL function can be used by guest systems BS2000/OSD as of V7.0.

- TCOM
The TCOM function is enabled on the guest systems on which TCOM is to be active.
- IOLVM (S servers)
Less important but I/O-intensive guest systems can hinder other, more important guest systems. This can occur when the I/O-intensive guest systems execute I/Os on the same (logical) device. It also occurs when I/Os are executed on different (logical) devices which are located on the same physical device or are connected over the same paths or can be reached via the same ports or are connected to the same channels.

IORM uses the IOLVM function to specifically slow down I/O operations of the user's own guest system if I/O resources (channel, port, path, disk) of a disk device that are used jointly exceed the specific I/O limit for the guest system. The I/O limit is specified as a percentage value of the average I/O throughput of the jointly used I/O resource.

The I/O limit for IOLVM is defined as the maximum I/O utilization of the VM in the `MAX-IO-UTILIZATION` operand in `/CREATE-VM` or `/MODIFY-VM-ATTRIBUTES`.

The IOLVM function is enabled on the guest systems on which IOLVM is to be active.

In the VM2000 information commands `/SHOW-VM-ATTRIBUTES` and `/SHOW-VM-RESOURCES` the `MAX-IO` column displays what value is set for the maximum I/O utilization of the VM. The display also indicates whether this value is effective, i.e. whether the IOLVM function has been enabled on the guest system.

Example

With the setting `MAX-IO-UTILIZATION=20`, IOLVM exerts control on the VM in the guest system when the I/O utilization exceeds 20% for a shared disk, for a path, for a port or for a channel.

6.4.9 Dynamic I/O configuration modification

6.4.9.1 Changing the configuration of channel devices via BS2000/OSD (S server)

On S servers, the input/output configuration of the channel peripherals can be changed during operation (dynamically). In other words, additions, changes or reductions can be made to the existing configuration without having to reload the system (IMPL). The changes are executed immediately on the active IOCF (IORSF) of the server and thus take effect immediately.

[SQ] No channels are available on SQ servers.

The system administrator starts dynamic I/O configuration modification with the command `/START-CONFIGURATION-UPDATE`. When this initiation is successfully concluded, the system administrator can change the I/O configuration dynamically with the commands `/ADD-IO-UNIT`, `/REMOVE-IO-UNIT` and `/MODIFY-IO-UNIT`.

Dynamic I/O configuration modification is concluded with the `/STOP-CONFIGURATION-UPDATE` command.

For more information on dynamic I/O configuration modification, which also applies to use under VM2000, see the “Introductory Guide to Systems Support” [2].

Dynamic I/O configuration modification via VM2000

Under VM2000 the commands for dynamic I/O configuration modification can only be entered in the monitor system. The `/START-CONFIGURATION-UPDATE` command is rejected if a guest system is in the startup phase. On the other hand, a guest system cannot be started (`/START-VM`) while dynamic I/O configuration modification is active.

When `/START-CONFIGURATION-UPDATE` is entered, message NKR0180 indicates how many guest systems are active and how many of them support dynamic I/O configuration modification.

The added devices are detected automatically by VM2000 and made known to all the active guest systems. The VM2000 administrator is informed of this by message VMS4065 and can then assign the new devices with `/ADD-VM-DEVICES`. The new devices must be attached in the guest system with `/ATTACH-DEVICE` before they can be operated in the current session. Guest systems that are started after dynamic I/O configuration modification recognize the new configuration automatically.

Changes to the configuration are automatically incorporated into the configuration descriptions of the monitor system and all guest systems.


The devices to be removed must not be assigned to a VM. The removed devices (`/REMOVE-IO-UNIT`) are also removed automatically from the device list of the active guest systems. The VM2000 administrator is informed of this by message `VMS4066`.

If a controller with several channel paths is to have a path changed with `/MODIFY-IO-UNIT`, the relevant path must be removed from **all** guest systems beforehand with `/REMOVE-DEVICE-CONNECTION`. This removal is facilitated by global reconfiguration in monitor system (see [page 263](#)).

The information commands of VM2000 and of the device management facility in BS2000/OSD provide information on the current configuration.

6.4.9.2 Changing the configuration of devices via X2000

The configuration of bus and FC peripheral devices managed on SQ servers by X2000 can be changed during ongoing VM2000 operation.

 The change is made via the SQ manager, see the manual “SQ Series Business Servers, Operation and Administration” [18].

1. Adding new devices

When it has been physically connected, the new device is defined and parameterized. The new device is reported by X2000 and is automatically recognized by VM2000, the monitor system and all guest systems.

The VM2000 administrator is informed of this with the message `VMS4065`.

The VM2000 administrator can then assign the new device to a VM, e.g. with `/ADD-VM-DEVICES`.

2. Changing device properties

When use of the device concerned has been prohibited in all guest systems by detaching it with `/DETACH-DEVICE`, the device properties can be changed. In this case the device must be removed from the VM only if its mnemonic device name changes. When the device properties have been changed, the device must be reattached to the guest system with `/ATTACH-DEVICE`.

3. Removing devices

The device concerned should be detached from all guest systems beforehand and removed from the VM with `/REMOVE-VM-DEVICES`. The relevant device is made invalid in the configuration of VM2000, in the monitor system and all guest systems.



The adding and removing of devices is immediately taken into account by the monitor system and all active guest systems.

6.4.9.3 Changing the configuration of devices in the disk storage system

When the configuration of a disk storage system is changed by the service, none of the devices involved may be assigned to a VM.

6.5 Backing up and restoring the VM2000 configuration

Backing up the current VM2000 configuration enables the VM2000 administrator to restore the backed-up configuration after the entire VM2000 system is (re)started, i.e. to restore the backed-up VM2000 configuration after VM2000 operation has been initialized or after all guest systems have been terminated and the monitor system has been restarted.

It is consequently no longer necessary to write or adapt a suitable procedure file with VM2000 commands to configure the VM2000 configuration.

The backup of a VM2000 configuration can also be used to restore the VM2000 configuration on another server (e.g. a backup server) on which no BS2000 guest systems run except the monitor system.

6.5.1 Backing up a VM2000 configuration

The VM2000 administrator can create the backup of a VM2000 configuration using the `/SAVE-VM-CONFIGURATION` command (see [page 525](#)).

In this case the components of the VM2000 configuration are stored in a backup file on the monitor system's home pubset, see [section "Backup file for a VM2000 configuration" on page 367](#).

Some components of the VM2000 configuration are not backed up, see [section "VM2000 components not backed up" on page 371](#).

A VM2000 configuration can only be backed up successfully if the prerequisites described below are satisfied at command entry time:

- No VM may be in the IN HOLD (WAIT) status
- All VMs in the RUNNING status must have reached SYSTEM READY
- Memory reconfiguration may not be in progress
- Live Migration (SQ servers) may not be in progress

Otherwise the command will be rejected.

When the prerequisites are fulfilled and there is an attached extra-CPU, the warning message `VMS4814` is issued. This is used to indicate that (chargeable) extra CPUs will be attached when the backup is read in.

6.5.2 Modifying the backup of a VM2000 configuration

The backup file in which a VM2000 configuration is stored can be edited using conventional editors. However, this is not recommended owing to the complexity of the file.

If you nevertheless want to edit the file, the directory with the start and end tags must be retained with the value ranges it supports. See also the [section “Structure of the contents” on page 367](#).

The dependencies which can exist between components of the VM2000 configuration must also be borne in mind.

Errors and inconsistencies in editing can result in it not being possible to execute `/RESTORE-VM-CONFIGURATION` (or only partially).

6.5.3 Restoring a VM2000 configuration

The VM2000 administrator can restore a VM2000 configuration using the `/RESTORE-VM-CONFIGURATION` command (see [page 521](#)).

In this case the components of the VM2000 configuration are restored from a backup file which was created beforehand on the monitor system's home pubset with `/SAVE-VM-CONFIGURATION`.

Components of the VM2000 configuration which were not backed up (see [section “VM2000 components not backed up” on page 371](#)) must, if necessary, be restored manually.

A VM2000 configuration can only be restored successfully if the prerequisites described below are satisfied at command entry time.

If one of the prerequisites is not fulfilled, the command is rejected or continued. The consequences of continuing the command are described in the various prerequisites.

Prerequisites:

- The current VM2000 version must be greater (younger) than or equal to the backed-up VM2000 version

If the current VM2000 version is less (older) than the backed-up VM2000 version, the backed-up VM2000 configuration could contain components which are not known in the current (older) VM2000 version.

- The current real main memory must be at least as large as at the time of the backup
Otherwise the warning message VMS4811 is output and processing is continued. If during further processing a VM cannot be configured because of a lack of main memory, the message VMS4803 is output for this VM.
- Prerequisites for the real CPUs:
 - The number of operable normal CPUs (ATT or DET status) must be at least as high as at the time of the backup
Otherwise two cases are distinguished:
 1. If the backup contains not only the CPU pool *STDP00L but also another VM2000 CPU pool, the command is rejected with the message VMS4812.
 2. If in addition to the CPU pool *STDP00L the backup contains no other VM2000 CPU pool, the warning message VMS4813 is output and processing is continued. If during further processing a VM cannot be configured again because the multiprocessor level is too high, the message VMS4803 is output for this VM.
 - The number of operable extra CPUs must be at least as high as at the time of the backup.
Otherwise two cases are distinguished:
 1. If the backup contains not only the CPU pool *STDP00L but also another VM2000 CPU pool, the command is rejected with the message VMS4812.
 2. If in addition to the CPU pool *STDP00L the backup contains no other VM2000 CPU pool, the warning message VMS4813 is output and processing is continued.
- Only the static CPU pools may exist
- No VM groups may exist
- Only the standard assignment set may exist
- No BS2000 VMs except the monitor VM may be configured on the system

- Prerequisites for the current monitor VM:
 - Memory reconfiguration may not be in progress
 - In the backed-up configuration the VM name of the current monitor VM is not assigned to any other VM than the monitor VM
Otherwise the warning message VMS4816 is output and processing is continued. If in the course of further processing a VM cannot be configured because of the VM name, message VMS4803 is output for this VM.
 - On S servers the location of the current monitor VM may not hinder the configuration of a backed-up VM (the location of the current monitor VM can change as a result of the changed size of the hypervisor)
Otherwise the warning message VMS4816 is output and processing is continued. If in the course of further processing a VM cannot be configured because of the location of the monitor VM, the message VMS4803 is output for this VM.
 - The main memory size of the current monitor VM may not be greater than the backed-up main memory size
On SQ servers the maximum main memory size of the current monitor VM may also not be greater than the backed-up maximum main memory size.
Otherwise the warning message VMS4816 is output and processing is continued. If in the course of further processing a VM cannot be configured because of the main memory size or the maximum main memory size of the monitor VM, the message VMS4803 is output for this VM.
 - The current explicit device assignments of the monitor VM must be a subset of the backed-up explicit device assignments
This prerequisite is not checked. If in the course of further processing the assignment of a device to a VM cannot be set up because of the assignment of this device to the current monitor VM, the message VMS4803 is output for this device.
- All devices which are assigned to a VM or an assignment set in the backed-up VM2000 configuration must be available to VM2000 again in the current system
This prerequisite is not checked. If in the course of further processing an assignment cannot be set up because a device is missing, the message VMS4803 is output for this device.

6.5.4 Backup file for a VM2000 configuration

When `/SAVE-VM-CONFIGURATION` is processed successfully, the current VM2000 configuration is stored in a backup file on the monitor system's home pubset. The file has the format of an XML document.

6.5.4.1 Storage location

The VM2000 administrator can define the name of the backup file explicitly.

If the backup file is to be assigned the default name or if the file name is specified without a user ID, the file name is completed in accordance with the caller's runtime environment:

- Command entry from a user task

The file name is completed with the caller's user ID. It is permissible to create the backup file under a foreign user ID if the caller has the TSOS privilege.
- Command entry from a KVP console or a logical console
 - Without operator LOGON

The file name is completed with the TSOS user ID. This ID is also the only one which may be specified in the file name.
 - With operator LOGON

The file name is completed with the operator user ID. It is permissible to create the backup file under a foreign user ID if the operator user ID has the TSOS privilege.
- Command entry via `$VMCONS`

The file name is completed with the user ID which is entered in the VM2000 parameter `VMCONS-OPERID` in the parameter file for the VM2-MON subsystem (see [page 49](#)). It is permissible to create the backup file under a foreign user ID if this user ID has the TSOS privilege.

6.5.4.2 Structure of the contents

The backup file has the format of an XML document. The XML document presents a tree structure. Every XML element begins with a **start tag** with the format `2<name>` and ends with the associated **end tag** `</name>` where *name* is an identifier for the tag.

The table below shows the structure of the VM2000 backup file with the meanings of the backed-up VM2000 components and the possible values. The end tags of the "pages" are not shown in order to keep the display concise.

XML tag	Meaning	Possible values
<vm2000_config>	Start of VM2000 configuration data	
<global_data> <vm2000_version> <max_num_vms> <max_num_vm_groups> <mon_vm_index> <real_mem_size> <new_dev_assign_set> <num_devices> </global_data>	Start of global data Version of VM2000 Maximum number of VMs Maximum number of VM groups VM index of the monitor VM Size of the real main memory (MB) Assignment set for newly added devices Number of devices End of global data	<product-version> <integer 1..99> <integer 1..50> <integer 1..99> <integer 1.. 1048576 *STDSET / <name 1..8> <integer 1..32767>
<cpu_pools> <cpu_pool_name> ... </cpu_pools>	Start of VM2000 CPU pools Name of the first CPU pool Further CPU pools (as before) End of VM2000 CPU pools	*STDPOOL / <name 1..8>
<real_cpus> <real_cpu> <cpu_id> <cpu_attr> <cpu_state> <cpu_pool_name> </real_cpu> ... </real_cpus>	Start of real CPUs First real CPU CPU identifier Type of CPU State of the CPU Name of the CPU's CPU pool End of first real CPU Further real CPUs (structure as before) End of real CPUs	<x-text 1..2> EXTRA / NORMAL / SPARE ATT / DET / ERR / OFFLINE / WAIT-FOR-MAINT *STDPOOL / <name 1..8>
<vm_groups> <vm_group> <vm_group_name> <cpu_quota> <max_cpu_util> <cpu_pool_name> </vm_group> ... </vm_groups>	Start of VM groups First VM group Name of the VM group CPU quota of the VM group Max. CPU utilization of the VM group Name of the VM group's CPU pool End of first VM group Further VM groups (structure as before) End of VM groups	<name 1..8> <integer 1..9999> <integer 1..10000> *STDPOOL / <name 1..8>
<assign_sets> <assign_set_name> ... </assign_sets>	Start of assignment sets Name of the first assignment set Further assignment sets (as before) End of assignment set	*STDSET / <name 1..8>

XML tag	Meaning	Possible values
<pre> <vms> <vm> <vm_index> <vm_name> <vm_group_name> <password> <mem_size> <min_mem_size> <max_mem_size> <mp_grade> <cpu_quota> <max_cpu_util> <location> <cpu_pool_name> <max_io_util> <admin_privileg> <vm_active_idle> <io_reset> <io_priority> <auto_snap_assign> <assign_by_guest> <assign_set_usage> <assign_sets> <assign_set_name> . . . </assign_sets> <auto_start> <ipl_unit> <skp_mc_out> <skp_mc_in> </vm> . . . </vms> </pre>	<pre> Start of BS2000 VMs First BS2000 VM VM index VM name Name of the VM's VM group Password of the VM VM's main memory size Min. main memory size of the VM Max. main memory size of the VM Multiprocessor level of the VM CPU quota of the VM Max. CPU utilization of the VM Location of the VM in the main memory (MB) Name of the VM's CPU pool Max. I/O utilization of the VM VM administrator's range of commands Control of the VM via real CPU IO-RESET privilege of the VM IO-PRIORITY privilege of the VM AUTO-SNAP-ASSIGNMENT privilege of VM ASSIGN-BY-GUEST privilege of the VM Type of specification for assignment sets Assignment sets of the VM Name of the VM's first assignment set Further assignment sets (as before) End of the VM's assignment sets Display of automatic start of the VM IPL device for the VM start Output path of KVP console when VM starts Input path of KVP console when VM starts End of first BS2000 VM Further VMs (structure as before) End of BS2000 VMs </pre>	<pre> <integer 1..99> <name 1..8> <name 1..8> <encrypted text 8..8> <integer 1..1048576> <integer 1..1048576> <integer 1..1048576> <integer 1..32> <integer 1..9999> <integer 1..10000> <x-text 1..8> *STDPOOL / <name 1..8> <integer 1..100> *MAXIMUM / *MINIMUM / *STANDARD *AT-DEDICATED-CPU / *NO *NO / *YES *NO / *YES *NO / *YES *ALL / *EXCEPT / *LIST *STDSET / <name 1..8> *NO / *YES <alphanum-name 2..2> / <x-text 4..4> <alphanum-name 2..2> <alphanum-name 2..2> </pre>

XML tag	Meaning	Possible values
<pre> <devices> <device> <mn> <pubres_attr_ind> <assign_by_guest> <assign_set_name> <usage_type> <explicit_sharers> <explicit_sharer> <vm_index> <pubres_assign_ind> </explicit_sharer> . . . </explicit_sharers> </device> . . . </devices> </pre>	<pre> Start of devices First device Mnemonic device name PUBRES-FOR-ATTRIBUTES attribute ASSIGN-BY-GUEST attribute Name of the device's assignment set Usage mode of the device Sharing VMs First sharing VM VM index of the VM PUBRES-FOR-ASSIGNMENT attribute End of first sharing VM Further VMs (structure as before) End of sharing VMs End of first device Further devices (structure as before) End of devices </pre>	<pre> <alphanum-name 2..2> / <x-text 4..4> *NO / *YES *NO / *YES *STDSET / <name 1..8> EXCL / SHARED / VC <integer 1..99> *NO / *YES </pre>
<pre> </vm2000_config> </pre>	<pre> End of VM2000 configuration data </pre>	

6.5.4.3 VM2000 components not backed up

The following components of the VM2000 configuration are **not** backed up:

- Passwords of \$VMCONS
- MONJVs
- Configuration of the global storage
- Monitor VM:
 - The multiprocessor level, main memory sizes and password of the monitor VM are backed up, but they are not included when restoration takes place.
 - Trace level
- Configuration statuses of the VMs

The AUTO-START=*YES attribute is only backed up for a VM whose guest system was started with a KVP pair as main console and using INFORMATION-BYTE=*FAST/*AUTOMATIC. An automatic restart of the guest system is possible for such a VM after the VM2000 configuration has been restored.

- The following applies for device configuration:
 - The device assignments to a pubset are not part of the VM2000 configuration.
In other words when the pubset structure or the device attributes change after the backup, the devices concerned do not receive the current assignments and attributes for the pubset when restoration takes place, but those which were saved for the devices when the backup was performed.
 - The assignments to VMs and the assignment to an assignment set are backed up for each device.
If a device does not belong to the current configuration when restoration takes place, these assignments cannot be restored.

6.5.4.4 Example of a backup file

For the sake of clarity some outputs in the example below are abbreviated (". . .").

```
<vm2000_config>
  <global_data>
    <vm2000_version>V10.0A00</vm2000_version>
    <max_num_vms>15</max_num_vms>
    <max_num_vm_groups>8</max_num_vm_groups>
    <mon_vm_index>1</mon_vm_index>
    <real_mem_size>8158</real_mem_size>
    <new_dev_assign_set>*STDSET</new_dev_assign_set>
    <num_devices>20061</num_devices>
  </global_data>
  <cpu_pools>
    <cpu_pool_name>*STDPOOL</cpu_pool_name>
    <cpu_pool_name>USRCPOOL</cpu_pool_name>
  </cpu_pools>
  <real_cpus>
    <real_cpu>
      <cpu_id>00</cpu_id>
      <cpu_state>ATT</cpu_state>
      <cpu_attr>NORMAL</cpu_attr>
      <cpu_pool_name>*STDPOOL</cpu_pool_name>
    </real_cpu>
    <real_cpu>
      <cpu_id>01</cpu_id>
      <cpu_state>ATT</cpu_state>
      <cpu_attr>NORMAL</cpu_attr>
      <cpu_pool_name>USRCPOOL</cpu_pool_name>
    </real_cpu>
    <real_cpu>
      <cpu_id>02</cpu_id>
      <cpu_state>DET</cpu_state>
      <cpu_attr>SPARE</cpu_attr>
    </real_cpu>
  </real_cpus>
```

```
<vm_groups>
  <vm_group>
    <vm_group_name>USRGROUP</vm_group_name>
    <cpu_quota>2500</cpu_quota>
    <max_cpu_util>2500</max_cpu_util>
    <cpu_pool_name>USRCPOOL</cpu_pool_name>
  </vm_group>
  <vm_group>
    <vm_group_name>MONGROUP</vm_group_name>
    <cpu_quota>5000</cpu_quota>
    <max_cpu_util>10000</max_cpu_util>
    <cpu_pool_name>*STDPOOL</cpu_pool_name>
  </vm_group>
</vm_groups>
<assign_sets>
  <assign_set_name>*STDSET</assign_set_name>
  <assign_set_name>USRASSET</assign_set_name>
  <assign_set_name>MNEASSET</assign_set_name>
</assign_sets>
<vms>
  <vm>
    <vm_index>1</vm_index>
    <vm_name>MONITOR</vm_name>
    <vm_group_name>MONGROUP</vm_group_name>
    <mem_size>512</mem_size>
    <min_mem_size>256</min_mem_size>
    <max_mem_size>8158</max_mem_size>
    <location>00000020</location>
    <mp_grade>2</mp_grade>
    <cpu_quota>100</cpu_quota>
    <max_cpu_util>9900</max_cpu_util>
    <cpu_pool_name>*STDPOOL</cpu_pool_name>
    <max_io_util>100</max_io_util>
    <admin_privileg>*STANDARD</admin_privileg>
    <vm_active_idle>*NO</vm_active_idle>
    <io_reset>*YES</io_reset>
    <io_priority>*NO</io_priority>
    <auto_snap_assign>*YES</auto_snap_assign>
    <assign_by_guest>*YES</assign_by_guest>
    <assign_set_usage>*LIST</assign_set_usage>
    <assign_sets>
      <assign_set_name>*STDSET</assign_set_name>
    </assign_sets>
    <auto_start>*NO</auto_start>
  </vm>
</vms>
```

```

<vm>
  <vm_index>2</vm_index>
  <vm_name>G4IVU</vm_name>
  <mem_size>256</mem_size>
  <min_mem_size>256</min_mem_size>
  <max_mem_size>8158</max_mem_size>
  <location>00000220</location>
  <mp_grade>2</mp_grade>
  <cpu_quota>2500</cpu_quota>
  <max_cpu_util>10000</max_cpu_util>
  <cpu_pool_name>USRCPOOL</cpu_pool_name>
  <max_io_util>100</max_io_util>
  <admin_privilege>*STANDARD</admin_privilege>
  <vm_active_idle>*NO</vm_active_idle>
  <io_reset>*NO</io_reset>
  <io_priority>*NO</io_priority>
  <auto_snap_assign>*YES</auto_snap_assign>
  <assign_by_guest>*YES</assign_by_guest>
  <assign_set_usage>*ALL</assign_set_usage>
  <auto_start>*YES</auto_start>
  <ipl_unit>E024</ipl_unit>
  <skp_mc_out>CH</skp_mc_out>
  <skp_mc_in>CI</skp_mc_in>
</vm>
. . .
</vms>
<devices>
  <device>
    <mn>CR</mn>
  </device>
  <device>
    <mn>CU</mn>
  </device>
  <device>
    <mn>CB</mn>
  </device>
  <device>
    <mn>CC</mn>
    <usage_type>VC</usage_type>
    <explicit_sharers>
      <explicit_sharer>
        <vm_index>2</vm_index>
      </explicit_sharer>
    </explicit_sharers>
  </device>
. . .

```

```
<device>
  <mn>E02F</mn>
  <assign_by_guest>*YES</assign_by_guest>
  <assign_set_name>*STDSET</assign_set_name>
  <usage_type>SHARED</usage_type>
  <explicit_sharers>
    <explicit_sharer>
      <vm_index>1</vm_index>
    </explicit_sharer>
    <explicit_sharer>
      <vm_index>3</vm_index>
    </explicit_sharer>
    <explicit_sharer>
      <vm_index>4</vm_index>
    </explicit_sharer>
    <explicit_sharer>
      <vm_index>5</vm_index>
    </explicit_sharer>
  </explicit_sharers>
</device>
. . .
<device>
  <mn>E024</mn>
  <pubres_attr_ind>*YES</pubres_attr_ind>
  <assign_by_guest>*YES</assign_by_guest>
  <assign_set_name>USRASSET</assign_set_name>
  <usage_type>EXCL</usage_type>
  <explicit_sharers>
    <explicit_sharer>
      <vm_index>2</vm_index>
      <pubres_assign_ind>*YES</pubres_assign_ind>
    </explicit_sharer>
  </explicit_sharers>
</device>
. . .
</devices>
</vm2000_config>
```

6.6 VM2000 in the high-availability cluster (SQ servers)

Two Server Units on different SQ servers can be combined to form a high-availability cluster (**HA cluster** for short) using X2000 V5.3 or higher.

In the HA cluster VMs in a failed Server Unit can be taken over by the other Server Unit, i.e. they are restarted on the other Server Unit.

VMs for which a failover by the other Server Unit is planned are identified via the SQ manager with the "**High-Availability VM**" attribute.

In an HA cluster virtual machines which run under VM2000 on a Server Unit can also be migrated to the other Server Unit while the BS2000 guest system is running. This is referred to as "**Live Migration**".

Operation and control of Server Units, HA clusters and Live Migration is performed on the SQ servers involved using the SQ manager. Basic information on this subject is provided in the manual "SQ Series Business Servers, Operation and Administration" [18].

This section describes the information functions of VM2000 which the VM2000 administrator can use to obtain information on an HA cluster with his/her VMs and devices. It also describes the general conditions and information outputs for Live Migration of a VM under VM2000 in the HA cluster.

6.6.1 VM2000 information on the VMs in the HA cluster

The VM2000 administrator can use `/SHOW-VM-RESOURCES INFORMATION=*CONFIGURATION` to ascertain whether his/her VM2000 system is operated on a Server Unit in an HA cluster.

When VM2000 is operated on a Server Unit in an HA cluster, the VM2000 administrator can obtain information on the configuration of all BS2000 VMs in this HA cluster with `/SHOW-VM-RESOURCES INFORMATION=*HA-CLUSTER`.

He/She can also use `/SHOW-VM-DEVICE-STATUS INFORMATION=*HA-CLUSTER` to obtain information on the device configuration of the BS2000 VMs in this HA cluster.

Conflicts in the HA cluster

Conflicts can occur in an HA cluster which would prevent a VM starting up after it has been migrated from one Server Unit to the other (e.g. after the Server Unit has failed). VM2000 can normally not prevent these conflicts from arising.



Exception

When configuring a VM, VM2000 uses `/CREATE-VM` to check whether the new VM name is unique in the HA cluster. To do this the name of the new VM is compared with all names of BS2000 VMs and XenVMs known in the HA cluster.

When VM2000 detects a conflict while processing one of the information commands for an HA cluster, this conflict is indicated specially in the output of the information command. The VM2000 administrator can then respond to this appropriately.

Taking over a VM in the HA cluster

When a Server Unit in the HA cluster fails and a BS2000 VM is taken over by the other Server Unit in the HA cluster, the message `VMS6020` (VM configured by the system administration) is output in the monitor system of the Server Unit which takes over the VM. The message `VMS4005` (device assigned by the system administration) is output for each device which is automatically assigned explicitly to the VM on the new Server Unit.

6.6.2 General conditions for Live Migration under VM2000

To permit Live Migration of a VM in an HA cluster, the Server Units of the SQ server must fulfill certain hardware conditions. The Server Units must be located in the same HA cluster, see the manual “SQ Series Business Servers, Operation and Administration” [18].

Under VM2000, too, no actions may be initiated or modifications performed for a VM which is migrated. Consequently some commands are rejected or only partially executed:

- The following VM2000 commands for the VM concerned are rejected with the message VMS3042:
 - /ADD-VM-DEVICES, REMOVE-VM-DEVICES **and** /SWITCH-VM-DEVICES (FORCE=*NO)
 - /ASSIGN-VM-TO-CPU-POOL
 - /DELETE-VM (CHECK-VM-STATE=*YES)
 - /EXTEND-VM-MEMORY, /REDUCE-VM-MEMORY
 - /HOLD-VM
 - /MODIFY-VM-ATTRIBUTES
 - /RESET-VM
 - /SHUTDOWN-VM
 - /START-VM (CHECK-VM-STATE=*YES)
- The following VM2000 commands are always rejected during Live Migration with the message VMS3042:
 - /SAVE-VM-CONFIGURATION
 - /SHUTDOWN-VM VM-ID=*VM2000(...)
- The following VM2000 commands only partially executed during Live Migration and the message VMS3041 and command return code VMS9005 are output:
 - /DELETE-VM VM-ID=*ALL, CHECK-VM-STATE=*YES
 - /HOLD-VM VM-ID=*ALL
 - /SHUTDOWN-VM VM-ID=*VM-GUESTS

6.6.3 VM2000 information during Live Migration of a VM

A BS2000 VM which is migrated (Live Migration) has the VM2000 status `MIGRATING`. This status is displayed in the Server Unit handing over the VM in the output column `STATE` in the VM2000 commands `/SHOW-VM-ATTRIBUTES` and `/SHOW-VM-RESOURCES`.

In the case of Live Migration of a BS2000 VM, the assigned devices are removed from the VM on the Server Unit which is handing it over. The message `VMS4009` (device removed by the system administration) is output for each removed device on the monitor system of the Server Unit handing over the VM.

The device assignments are automatically restored on the receiving Server Unit. The message `VMS4005` (device assigned by the system administration) is output on the monitor system of the receiving Server Unit for each such device assignment.

When the VM is configured on the receiving Server Unit, the message `VMS6020` (VM configured by the system administration) is output on the monitor system of the Server Unit which hands over the VM.

When Live Migration of the VM has been concluded and the VM on the Server Unit which hands it over is terminated, the message `VMS6021` (VM terminated by the system administration) is output on the monitor system of the Server Unit handing over the VM.

6.7 Password protection

VM2000 has a twofold password protection system:

- Password protection for administering a VM. This password is defined when a VM is initialized and checked during dialog access to a VM (ADMIN dialog or VC dialog).
- Password protection for access to \$VMCONS. This password is defined in the parameter file of the VM2-MON subsystem and checked during connection setup to \$VMCONS.



Passwords consisting only of zeros (X'00..00') are ignored; the desired password is not defined (same effect as `PASSWORD=*NONE`).

Password protection for access to a VM

When initializing a VM (`/CREATE-VM`), the VM2000 administrator can use the `PASSWORD` operand to define a password which authorizes initiation of an administration dialog from a privileged user task or via \$VMCONS (ADMIN or VC dialog) with a VM. If a password is defined, it must be specified when a dialog is initiated with `/BEGIN-VM-DIALOG`.

The password can be changed or deleted by the VM2000 administrator by means of `/MODIFY-VM-ATTRIBUTES` during VM operation. The password remains valid until the VM is terminated (`/DELETE-VM`) or until the end of the current session (shutdown of the monitor system). In particular, it also remains valid after an automatic restart of the monitor system or of \$VMCONS.



On S servers the password for administering the monitor VM can be defined via the configuration file for the monitor VM (see [page 57](#)) and on SQ servers via the configuration of VM2000 in the SQ manager (see [page 62](#)).

Password protection for access to \$VMCONS

A password can be specified in the parameter file of the VM2-MON subsystem which grants authorization for establishing a connection to \$VMCONS (`VMCONS-PASSWORD`, see [page 49](#)). If a password has been declared, it must be given before a connection to \$VMCONS can be established.

The password can be changed during operation by means of `/MODIFY-VMCONS-PARAMETERS` (see [page 501](#)). The password remains valid until the end of the of the current session or until the command is entered again. It also remains valid after an automatic restart of \$VMCONS. However, it is no longer available after an automatic restart of the monitor system. Modifications for the next session must be made via the parameter file.

6.8 Accounting in VM2000

VM2000 supplies two types of accounting records:

- Accounting record for devices. This record contains job accounting data on devices assigned to a VM.
- Accounting record for a VM. This record contains VM-specific job accounting data.

Device accounting records are written at the following times:

- When implicitly or explicitly assigned devices are removed from a VM or implicitly assigned devices are released by the guest system

Devices are removed from a VM by VM2000 by:

- `/REMOVE-VM-DEVICES` or `/SWITCH-VM-DEVICES` (implicitly or explicitly assigned devices selectively)
- `/DELETE-VM` (all devices)
- `/SHUTDOWN`, abnormal system termination or restart in the guest system, `/SHUTDOWN-VM`, `/START-VM` or `/RESET-VM` (implicitly assigned devices, see [page 290](#))
- `/SHUTDOWN` in the monitor system (`/DELETE-VM` for all VMs)
- a restart in the monitor system, automatically or via SVP (implicitly assigned devices are released before `SYSTEM READY`, see [page 290](#))
- Live Migration of the VM (SQ servers)

The guest system on a VM releases implicitly assigned devices by means of:

- `/DETACH-DEVICE` in the guest system
- When `/MODIFY-VM-ATTRIBUTES` is specified and the same name is given for the VM as before and devices are assigned to the VM

An accounting record for a VM is written at the following times:

- The VM's main memory is reconfigured with `/EXTEND-VM-MEMORY` or `/REDUCE-VM-MEMORY`
- The assignment of the global memory to the VM is changed with `/ADD-VM-RESOURCES` or `/REMOVE-VM-RESOURCES`
- The GS complex configuration is changed (S servers)
- A VM is terminated with `/DELETE-VM`.
- The same VM name is assigned to the VM as before using `/MODIFY-VM-ATTRIBUTES`
- The monitor system is terminated with `/SHUTDOWN-VM VM-ID=*VM2000` or `/SHUTDOWN`
- Live Migration of the VM has taken place (SQ servers)



Interruption-free switching between to and from daylight savings time is supported by appropriate information in the accounting records.

If the user of the VM changes (without the VM being terminated and newly initialized), an accounting record can be created via the VM name with `/MODIFY-VM-ATTRIBUTES`. The new name must be the same as the old one.

The accounting records are written by the BS2000 accounting system in the monitor system (see the “Introductory Guide to Systems Support” [2]). The accounting system in the monitor system must therefore be active at all times.

The accounting records can be analyzed with the software product RAV (see the “RAV” manual [7]). In this context, VM names can be managed in the same way as user IDs (see “[Recommendations for the definition and use of VM names](#)” on [page 139](#)).

Other guest systems do not write VM2000 accounting records.

Structure of the accounting records

The accounting records are divided up into four sections:

- record description section
- ID section
- basic information
- variable information.

These record sections contain the field number, displacement, length and format of the data field.

The **field number** indicates the current sequence number within the record section.

The **displacement** is the distance of the data field from the start of the record section.

The **length** is the length of the data field in bytes.

The **format** is the format of the data field:

- A alphanumeric (including “\$”, “#” and “@”)
- B binary number
- C printable characters
- F BS2000 file name
- Z decimal number, unpacked (0...9)
- undefined

6.8.1 Accounting record for devices

Structure of the record description section

The record description section contains the record ID, the time of day as well as the length of the ID section and the basic information.

Field no.	Displ.	Length	Format	Meaning
1	00	4	A	Record ID 'VACD'
2	04	8	-/B	Time Stamp
3	0C	2	B	Length of ID section
4	0E	2	B	Length of basic information
5	10	4	-	Reserved

Structure of the ID section

The ID section contains the VM identification, i.e. the VM index and the VM name.

Field no.	Displ.	Length	Format	Meaning
1	00	3	Z	VM index
2	03	3	-	Reserved
3	06	8	A	VM name
4	0E	2	-	Reserved

Structure of the basic information

The basic information contains the information regarding summer time/winter time, date and time of the release of the devices.

Field no.	Displ.	Length	Format	Meaning
1	00	6	Z	Date of release of devices, 1st part (format: yymmdd)
2	06	6	Z	Time of release of devices (format:hmmss) (format: hhmmss)
3	0C	2	Z	Date of release of devices, 2nd section: century of release (format: yy)
4	0E	1	C	Time of year of the release of the devices (S=summer time, W=winter time)
5	0F	3	-	Reserved

Structure of the variable information

The variable information contains

- the individual devices
- summer time/winter time, date and time of assignment to the VM
- device type in the monitor system
- mnemonic device name in the monitor system

Field no.	Displ.	Length	Format	Meaning
1	00	2	B	Number of extensions = 1
2	02	2	B	Distance between record extension for devices and start of record

If the displacement value is 0, the corresponding record extension is not specified.

Record extension for devices

Field no.	Displ.	Length	Format	Meaning
1	00	2	A	Extension identification = 'DV'
2	02	1	B	Number of devices = number of extensions of type 'DV'
3	03	1	B	Length of extension of type 'DV'
4	04	2	-	Reserved
5	06	6	Z	Date of assignment of device, 1st part (format: yymmdd)
6	0C	6	Z	Time of assignment of device (format: hhmmss)
7	12	8	C	Device type
8	1A	4	C	Mnemonic device name
9	1E	2	Z	Date of assignment of device, 2nd part: century of assignment (format: yy)
10	20	1	C	Time of year of the assignment of the devices (S=summer time, W=winter time)
(n-1) times fields 5 to 10 in accord. with number of devices in field 2				see fields 5 to 10

6.8.2 Accounting record for a VM

Structure of the record description section

The record description section contains the record ID, the time of day and the length of the ID section and the basic information.

Field no.	Displ.	Length	Format	Meaning
1	00	4	A	Record ID 'VACM'
2	04	8	-/B	Time Stamp
3	0C	2	B	Length of ID section
4	0E	2	B	Length of basic information
5	10	4	-	Reserved

Structure of the ID section

The ID section contains the VM identification, i.e. the VM index and the VM name.

Field no.	Displ.	Length	Format	Meaning
1	00	3	Z	VM index
2	03	3	-	Reserved
3	06	8	A	VM name
4	0E	2	-	Reserved

Structure of the basic information

The basic information contains

- start of the accounting period (summer time/winter time, date and time)
- end of the accounting period (summer time/winter time, date and time)
- CPU time used
- main memory size (see [page 270](#))
- size of global storage (see [page 277](#))

Field no.	Displ.	Length	Format	Meaning
1	00	6	Z	end of the accounting period (date) (format: yymmdd)
2	06	6	Z	end of the accounting period (time) (format: hhmmss)
3	0C	6	Z	start of the accounting period (date) (format: yymmdd)
4	12	6	Z	start of the accounting period (time) (format: hhmmss)
5	18	4	B	CPU time used (seconds) ¹
6	1C	4	B	CPU time used (microseconds) ¹
7	20	2	B	Main memory size (in Mbytes) ²
8	22	2	B	Size of global storage (in Mbytes) ²
9	24	2	Z	end of the accounting period (century) (format: yy)
10	26	2	Z	start of the accounting period (century) (format: yy)
11	28	1	C	end of the accounting period (season) (S=summer time, W=winter time)
12	29	1	C	start of the accounting period (season) (S=summer time, W=winter time)
13	2A	8	-	Reserved
14	32	2	B	Main memory size or VMs from 64 Gbytes of main memory (in 64 Gbytes). The size modulo 64 Gbytes is output in field 7.
15	34	2	B	Global storage size for global storages from 64 Gbytes (in 64 Gbytes). The size modulo 64 Gbytes is output in field 8.

¹ See the description of VM-ACTIVE-IDLE on [page 147](#)

² The size modulo 64 Gbytes is output here for VMs with a main or global storage size \geq 64 Gbytes. The size in multiples of 64 Gbytes is output in field 14 or 15.

Structure of the variable section

Field no.	Displ.	Length	Format	Meaning
1	00	2	B	Length of variable section (=0)

The VM accounting record has no variable section.

6.9 Hardware error logging under VM2000

Hardware error logging (HEL) is a permanent procedure for logging hardware errors, e.g.:

- machine fault
- error caused by channel interrupt
- fault in operation in peripheral devices

These errors are logged in the so called “HEL records” of the HEL file.

The following information is also stored in the HEL file:

- statistical information from peripheral devices
- information on debugging and diagnostics programs.

HEL entries help to trace problems and are not required for the operation of VM2000. For more information on working with HEL, refer to the “ELSA” manual [6].

The errors and events resulting in entries in the HEL file are reported in native mode as under VM2000 by the individual guest systems to the local error logging system (ELS) of BS2000 and stored by ELS in a local HEL file of the guest system. Under VM2000 this is referred to as “VM-local hardware error logging (HEL)”.

The same physical event can affect several guest systems. Each guest system affected notes the event in its HEL file.

Example

When an error occurs in connection with a channel interrupt or I/O timeout, the error is noted and logged in each guest system affected.

Global hardware error logging (HEL)

All HEL records encountered under VM2000 on a BS2000/OSD server but which are in different virtual machines are stored in a central file in the monitor system (global VM HEL file) as well as in the local VM HEL file.

This “VM-global hardware error logging (VMHEL)” is carried out by VM2000 for all guest systems.

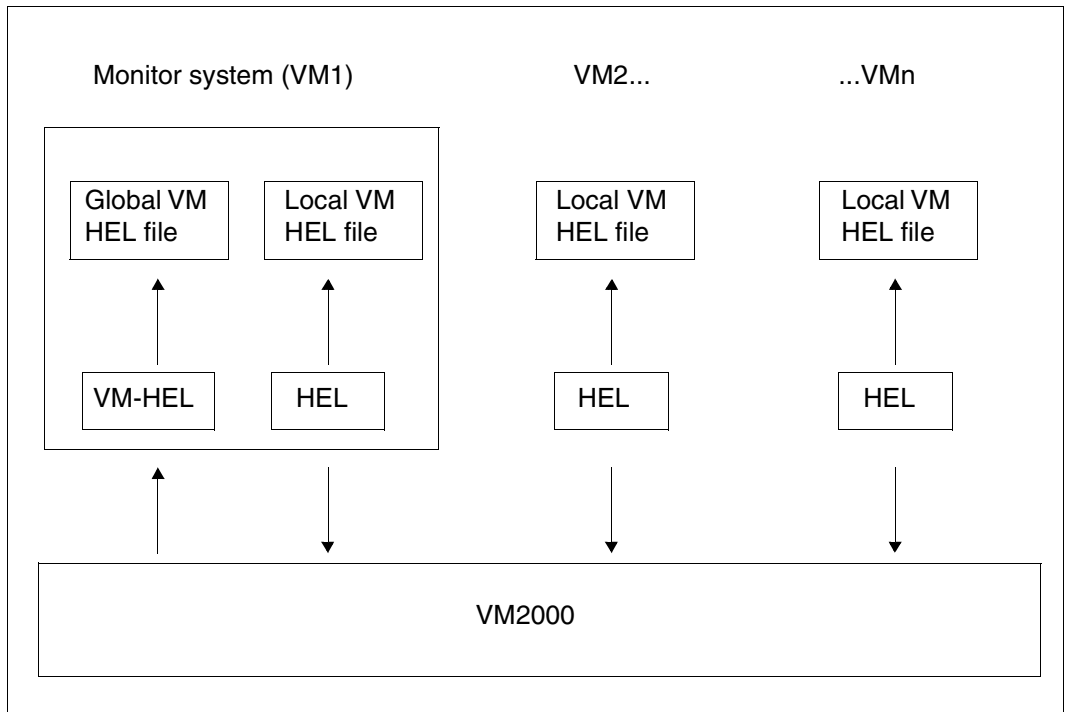


Figure 30: Concept of cross-VM hardware error logging under VM2000

Local hardware error logging (HEL) in every guest system, also that in the monitor system, processes only the local HEL file in the guest system and does not know the global HEL file.

VM-HEL processes the global VM HEL file and is performed exclusively in the monitor system. The records to be stored are supplied to the global VM-HEL by the local HELs of each VM.

All attributes and processing options of the local VM HEL file apply equally to the global VM HEL file, e.g. automatic exchange of the HEL file.

The layout of the global VM-HEL and local HEL records is identical (a detailed description of the analysis of these records can be found in the “ELSA” manual [6]).

The following control commands are available for the global VMHEL. The functions of these commands correspond to the specified control commands of the local HEL but apply to the global VMHEL or the global VMHEL file in the monitor system. These commands are only available under VM2000 in the monitor system.

VMHEL command (corresponding HEL command)	Meaning
CHANGE-VMHEL-FILE (CHANGE-HEL-FILE)	Closes current global (local) HEL file and opens a new file
MODIFY-VMHEL-CHECK (MODIFY-HEL-CHECK)	Controls global (local) threshold monitoring
MODIFY-VMHEL-LOGGING (MODIFY-HEL-LOGGING)	Controls the saving of the global (local) HEL records
MODIFY-VMHEL-TELESERVICE-ALARM (MODIFY-HEL-TELESERVICE-ALARM)	Specifies if and in what period of time the “Call-Teleservice” message is to be output when a global (local) error threshold is reached
SHOW-VMHEL-CHECK (SHOW-HEL-CHECK)	Shows the current settings for the global (local) threshold monitoring
SHOW-VMHEL-LOGGING (SHOW-HEL-LOGGING)	Requests information on the global (local) logging records
SHOW-VMHEL-STATUS (SHOW-HEL-STATUS)	Requests general information about the status of the global (local) hardware error logging
SHOW-VMHEL-TELESERVICE-ALARM (SHOW-HEL-TELESERVICE-ALARM)	Requests information about the set global (local) teleservice parameters
START-VMHEL-LOGGING (START-HEL-LOGGING)	Activates global (local) hardware error logging and opens global (local) HEL file
STOP-VMHEL-LOGGING (STOP-HEL-LOGGING)	Ends global (local) hardware error logging and closes global (local) HEL file



The commands require the system privilege HWMMAINT (HARDWARE-MAINTENANCE). The VMHEL commands have the same syntax as the HEL commands. The HEL commands are described in detail in the “Commands” manual [12].

The messages of the global VMHEL have the message code HLVnnnn. The meaning and response texts for a message can be inquired during ongoing operation using the BS2000 command /HELP-MSG-INFORMATION HLVnnnn.

6.10 Monitoring under VM2000

The following monitoring options are available under VM2000:

- monitoring under VM2000 with `/SHOW-VM-STATUS`
- monitoring in the guest systems or in the monitor system with the aid of the measurement monitor openSM2

The following restrictions govern both options:

- under VM2000, CPU times must not be calculated as the difference between two STCK values
- IDLE times cannot be monitored precisely in a guest system under VM2000

6.10.1 Measurement with `/SHOW-VM-STATUS`

`/SHOW-VM-STATUS` gives the VM2000 administrator an overview of the current measured values for active objects under VM2000, see [page 607](#). The command supplies measured values in the following categories once or periodically:

- Operand `INFORMATION=*TIMER/*ALL`:
Values for usage of CPU capacity for the real CPUs, the CPU pools, the VM groups, and the VMs.
- Operand `INFORMATION=*COUNTER/*ALL`:
Events which were executed by the VM2000 hypervisor for real or virtual CPUs.
- Operand `INFORMATION=*SCHEDULE/*ALL`:
Measured values for scheduling by the VM2000 hypervisor for real CPUs and VMs.

[SQ] The operand values `INFORMATION=*COUNTER/*SCHEDULE` are not available on SQ servers.

These values are output in each case as absolute values and also as percentages.

The one-off output of measured values (`PERIOD=*RECENT`) takes place synchronously and shows the measured values from the immediate past. One-off output can be called in parallel by several VM2000 administrators.

In the case of `/SHOW-VM-STATUS PERIOD=<integer>` you can specify the interval at which the measured values are to be output. You can set an interval of 5 seconds to 1 hour. It makes sense to set several minutes so that the time values provide meaningful information as regards the load imposed on the CPU by the VMs. Periodic output can only be used by one VM2000 administrator at a time.

6.10.2 Measurement with the openSM2 measurement monitor

The openSM2 (for short: SM2) measurement monitor can collect measurement values in the monitor system and in the guest systems. SM2 provides measured values for the guest system.

The scope of the VM2000 measured values (in the VM2000 report and in the SM2R1 reports 98 and 99) differs for the monitor system and the guest systems: in the monitor system the measured values of all VMs are output, in a guest system only the measured values of the VM on which the guest system is running.

The measurement methods and reports are described in the “openSM2” manual [8]. The peculiarities and interpretations of the measured values under VM2000 are described in the section entitled “SM2 together with VM2000”.



In the monitor system and in all guest systems, CPU pools and VM groups can also be seen in the SM2 outputs.

The following should also be noted for S servers:

- The Channel Subsystem Monitoring Facility is required in order to measure the device-specific operating times (monitoring program `SERVICETIME`). This firmware feature can only be allocated to **one** guest system at a time. This means that the first guest system to call this measurement function by means of the SM2 statement `//START-MEASUREMENT-PROGRAM TYPE=*SERVICETIME` during VM2000 operation occupies this firmware feature for the relevant VM.

If another guest system attempts to occupy this firmware feature when the Channel Subsystem Monitoring Facility is already allocated to a VM, SM2 issues a warning. In this case, SM2 can only record the `DEVICE QUEUE` and `REMAINING SERVICE TIME`.

The Channel Subsystem Monitoring Facility can be released again:

- by means of the SM2 statement `//STOP-MEASUREMENT-PROGRAM TYPE=*SERVICETIME`
- when the guest system is restarted on the VM
- by means of the VM2000 commands `/DELETE-VM`, `/RESET-VM` and `/START-VM`

The VM2000 administrator receives the current occupation status with `/SHOW-VM-RESOURCES INFORMATION=*STD/*ALL`. If the Channel Subsystem Monitoring Facility is occupied by a VM, message `VMS2035` is output as well.

- The Channel Subsystem Monitoring Facility is activated for exclusively used devices and for shareable devices that are only assigned to one VM (direct I/O). It is **not** activated for shareable devices that are assigned to more than one VM (indirect I/O). SM2 issues a warning message here. In this case, SM2 can only record the `DEVICE QUEUE` and `REMAINING SERVICE TIME`.

7 Error handling in VM2000

This chapter describes:

- the types of error which can occur during VM2000 operation
- the restart procedures provided in VM2000 and
- the diagnostic aids VM2000 provides.

Whenever an error occurs you should create diagnostic documentation and notify Customer Support.

7.1 Types of error in VM2000

The following types of error can occur during VM2000 operation:

- Errors in the carrier system (SQ servers)
 - Errors in the Xen hypervisor or in the carrier system 's kernel
 - Errors in the administration component of the carrier system
 - errors in the VM2000 Agent
- VM2000 errors
 - Errors during VM2000 startup
 - errors in the VM2000 hypervisor (S servers)
 - errors in the VM2000 monitor
 - errors in \$VMCONS
- Guest system errors
 - SETS in the BS2000 guest system
 - SETS in the monitor system
 - monitor system deadlock

The following sections list the individual error types, outlining the corresponding message text, the cause of the error and possible corrective measures.

7.1.1 Errors in the carrier system (SQ servers)

Errors in the Xen hypervisor or in the carrier system 's kernel

Form	Server operation is terminated abnormally
Cause	Software error in the Xen hypervisor or in the carrier system's kernel
Diagnostic materials	See the manual "SQ Series Business Servers, Operation and Administration" [18]
Response	The server unit is restarted with the current settings. VM2000 is initialized again. Virtual machines with a BS2000 guest system must be re-initialized.

Errors in the administration component of the carrier system

Form	Message VMS6001 in the monitor system.
Cause	Software error in the administration component of the carrier system
Diagnostic materials	Core dump (for Customer Support)
Response	<p>case a) The component is automatically restarted. VM2000 mode continues to run without hindrance.</p> <p>case b) The component cannot be automatically restarted. VM2000 operation continues. However, no VM2000 commands can be executed. BS2000 guest systems cannot issue any hypervisor requests; they receive the return code VMS6001. Notify Customer Support.</p>

errors in the VM2000 Agent

Form	Messages VMS6000 and VMS6003 in the monitor system.
Cause	Software error in the VM2000 Agent
Diagnostic materials	Core dump (for Customer Support)
Response	<p>case a) The VM2000 Agent is restarted automatically. VM2000 mode continues to run without hindrance.</p> <p>case b) The VM2000 Agent cannot be restarted automatically. VM2000 operation continues. However, no VM2000 commands can be executed. BS2000 guest systems cannot issue any hypervisor requests; they receive the return code VMS6000. Notify Customer Support.</p>

7.1.2 VM2000 errors

Errors during VM2000 startup

SQ On SQ servers the startup of the monitor system is continued even if an error occurs when initializing VM2000 operation.

Form	Message: VMS50xx... , STARTUP TERMINATED
Cause	Error during VM2000 startup, as described in message VMS50xx
Diagnostic materials	Console log/CONSLOG file; SLED of the overall system
Response	Eliminate the error and reload VM2000

Errors in the VM2000 hypervisor (S servers)

SQ On SQ servers the Xen hypervisor controls all virtual machines, see [page 394](#).

Abnormal VM2000 hypervisor termination also terminates the operation of all the guest systems and of VM2000. If an automatic restart in the monitor system is not set, VM2000 and the monitor system must be reloaded manually. If an automatic restart in the monitor system is set, VM2000 and the monitor system are automatically reloaded (see [page 401](#)). After the monitor system has started up, the VMs must be set up again and the guest systems restarted.

Form	Message: VMS00xx Exceptions: messages VMS0017, VMS0018, VMS0021, VMS0023 through VMS0027 and VMS0000
Cause	The VM2000 hypervisor recognizes a system error (described in message VMS00xx) and terminates VM2000.
Diagnostic materials	Console log/CONSLOG file; SLED of the system as a whole
Response	Reload VM2000/automatic restart

Exception: abnormal system termination in the monitor system

Form	Message: VMS0018
Cause	Abnormal system termination in the monitor system. A restart option is not set or the restart function has failed. At least one guest system is still active.
Diagnostic materials	Console log/CONSLOG file
Response	Shutdown for all guest systems or attempt to restart the monitor system via SVP.

Exception: error in a CPU

Form	Message: VMS0017 and VMS0021
Cause	The real CPU was detached implicitly due to a hardware or software error.
Diagnostic materials	Console log/CONSLOG file; automatic VMDUMP (see page 408)
Response	Respond to message VMS0021. Contact customer support. Check CPU and attach again, if applicable, if no hot spare CPU is available.
Form	Message: VMS0023 and VMS0024
Cause	A hot spare CPU was attached following CPU failure.
Diagnostic materials	None
Response	Respond to message VMS0024. Contact customer support. Repair defective CPU.

Exception: CPU pool reconfiguration after a CPU error

Form	Message: VMS0025, VMS0026 or VMS0027
Cause	Owing to a CPU error a CPU, a VM or a VM group had to be assigned to another CPU pool.
Diagnostic materials	None
Response	Check CPU pools and reorganize if applicable.

Exception: other errors

If any other errors that have not yet been described occur in the VM2000 hypervisor, an automatic VMDUMP is created and output. It should be forwarded to customer support for diagnosis.


A special message is not issued.

Errors in the VM2000 monitor or in \$VMCONS

Form	<p>case a) System dump of a system task (VM2M, VM2G, VM2C, VM2J, VM2P, VM2E)</p> <p>case b) Message issued on the BS2000 console of the monitor system or in the ADMIN dialog of the monitor VM VMS2014 VM2000 PROGRAM ERROR. MODULE '(&00)', LABEL '###(&01)', ERROR CODE '(&02)'</p> <p>Key:</p> <p>(&00): Name of the VM2000 module in which the error was detected</p> <p>###(&01): Flag in VM2000 module (&00)</p> <p>(&02): More detailed error information</p>
Cause	<p>case a) Program error in a system task</p> <p>case b) Internal inconsistency detected or serious VM2000 error</p>
Diagnostic materials	<p>case a) CONSLOG file of the monitor system; system dump</p> <p>case b) CONSLOG file of the monitor system; automatic VMDUMP (see page 408).</p>
Response	<p>Continue with VM2000 operation. The affected system task is continued from a restart point (see page 405). It may be necessary to terminate the affected VM by means of /DELETE-VM and to establish \$VMCONS connections again.</p>

7.1.3 Guest system errors

SETS in the BS2000 guest system

Form	<ul style="list-style-type: none"> – Message in the BS2000 guest system: 'crash-id: SETS; text' <p>and</p> <ul style="list-style-type: none"> – Message on the BS2000 console of the monitor system or in the ADMIN dialog of the VM2000 or VM administrator: VMS2033 'crash-id: SETS;text' FROM VM ((index),(name)) VIA SVP <p>where</p> <p>crash-id Crash ID of the module by which the system was terminated.</p> <p>index VM index</p> <p>name VM name</p>
Cause	Global system error in the guest system.
Diagnostic materials	<p>The SLED of the guest system can be created by means of /START-VM ... ,DIAGNOSTIC-IPL=*YES</p> <p> The diagnostic data of the VM2000 hypervisor is also contained in the SLED of the guest system (dump file section VM2HYPVS).</p>
Response	<ul style="list-style-type: none"> – VM2000 operation continues unaffected. – The guest system can be reloaded.



If an automatic restart has been defined in the guest system (see [page 402](#)), the main console (specified by means of the `MAIN-CONSOLE` operand in `/START-VM`) and the IPL device (specified by means of the `IPL-UNIT` operand in `/START-VM`) of this VM must still be assigned at the restart. Implicitly assigned devices are removed from the VM when the guest system is restarted.

See [page 290](#) for details on how to handle implicitly assigned devices at `/START-VM`.

The VM administrator is informed about the failure of the guest system, depending on the restart option in the guest system, by means of one of the following messages:

- VMS2051 (“Guest system on VM (...) down; reason: crash”)
- VMS2052 (“Guest system on VM (...) not ready. Restart has been initiated”)

The messages are also issued in the monitor system using routing code “9”.

SETS in the monitor system

After SETS in the monitor system, the guest systems remain operable and can be operated, for example, using BS2000 consoles, but not in a VC dialog. VM2000 and the virtual machines cannot, however, be addressed using VM2000 commands. Refer to [page 402](#) for details on automatic restarting of the monitor system.

If no automatic restart is set in the monitor system, then the following applies:

- S On S servers the guests systems can be shut down using /SHUTDOWN and VM2000 initialized again. Alternatively, the monitor system can be restarted (with SLED) via the SVP.
- SQ On SQ servers first the monitor system's SLED can be created and subsequently the monitor system can be restarted (see below and the manual "SQ Series Business Servers, Operation and Administration" [18]).
- i The guest systems are informed about the failure of the monitor system. One of the following messages is also displayed on the guest system BS2000 console, depending on the restart option in the monitor system:
 - NRTV001 ("Monitor system failed")
 - NRTV002 ("Monitor system not ready. Restart has been initiated")

Reaction of the monitor system to SETS (automatic restart deactivated):

Form when automatic restart is deactivated

Messages issued to the monitor system BS2000 console:
'crash-id: SETS; text'

and (in the case of S servers)

```
VMS0000 MONITOR SYSTEM TERMINATED. VM2000 TERMINATION
or
VMS0018 MONITOR SYSTEM TERMINATED.
VM2000 ADMINISTRATION IMPOSSIBLE
```

where `crash-id` is the crash ID of the module by which the system was terminated.

Cause

Global system error in the monitor system

Diagnostic materials

- S VMS0000: SLED of the system as a whole
VMS0018: SLED of the system as a whole or the monitor system
- SQ SLED of the monitor system (via SVP functions of the assigned KVP console)

Response

- S VMS0000: Initialize VM2000 operation again
VMS0018: Restart via SVP, see [page 403](#), if possible; otherwise, terminate all guest systems and initialize VM2000 mode again
- SQ Restart monitor system (via SQ manager or SVP functions of the assigned KVP console)

Monitor system deadlock

Problems may occur in the monitor system which prevent a `/SHUTDOWN` command from being issued, but do not cause a crash (monitor system is hung up or endless loop occurs, `UCON BUSY`, no entry possible).

- S On S servers the monitor system can be restarted (with SLED) via the SVP, see [page 403](#).
- SQ On SQ servers first the monitor system's SLED can be created and subsequently the monitor system can be restarted (via the SQ manager or the SVP functions of the assigned KVP console, see the manual "SQ Series Business Servers, Operation and Administration" [18]).

During this process, the guest systems will remain operable, but cannot be addressed using VM2000 commands or via VC dialog when the dump is being generated or while the monitor system is being restarted automatically.

The guest systems are only informed about the system standstill in the monitor system after start or restart of the monitor system has been initiated. Message `NRTV002` ("Monitor system not ready. Restart has been initiated") is output on the guest system BS2000 console.

- i If due to an I/O problem in the monitor system only BCAM is restarted, no messages are generated in the guest systems.

7.2 Restart routines in VM2000

Restart routines are executed by VM2000 following abnormal VM2000 hypervisor termination (on S servers), abnormal VM2000 Agent termination (SQ servers), after SETS in the monitor system and following abnormal task termination in one of the system tasks of VM2000.

7.2.1 Automatic restart following termination of the VM2000 hypervisor (S servers)

[SQ] On SQ servers the Xen hypervisor controls all virtual machines, see [page 394](#).

Abnormal VM2000 hypervisor termination also terminates the operation of all the guest systems and of VM2000.

The automatic restart system function in the monitor system enables VM2000 and the monitor system to be automatically started again after abnormal VM2000 hypervisor termination.

This involves carrying out all the preparations necessary for an automatic restart in BS2000 mode without VM2000 (refer to the list in the [section “Automatic restart after SETS in the monitor system” on page 402](#)). The following should also be noted:

- The setting `DUMP=*SNAP` in `/SET-RESTART-OPTIONS` has the same effect as `DUMP=*SLED`, i.e. a complete memory dump is created.
- The runtimes for VM2000 and the monitor system differ from each other and are independent of one another. The runtime of VM2000 begins with `VM2000 READY` and the monitor system runtime begins with `SYSTEM READY` (it can be shorter than the VM2000 runtime). The time `UPTIME` specified for `/SET-RESTART-OPTIONS` applies to both runtimes.

After a restart of the monitor system (or of VM2000), automatic restart is deactivated (`/SET-RESTART-OPTIONS MODE=*OFF`). The restart option must be reset again with `/SET-RESTART-OPTIONS` and then applies to both the restart after SETS in the monitor system and the automatic restart following VM2000 hypervisor termination.

Special case: If abnormal VM2000 hypervisor termination occurs during a restart of the monitor system (not VM2000), VM2000 and the monitor system are restarted.

After the monitor system started up, the VMs must be set up again by the VM2000 administrator and the guest systems must be restarted. If, due to system failure, applications were relocated to a standby server using HIPLEX AF, this must be taken into account when the VMs are set up again.

`/SHUTDOWN` with a restart in the monitor system does **not** result in a VM2000 restart.



When VM2000 restarts, the VM2000 settings of the startup parameter service are evaluated, **not** the most recent valid VM2000 settings.

7.2.2 Automatic restart following termination of the VM2000 Agent

On SQ servers the VM2000 Agent is automatically restarted after terminating abnormally. A core dump is also created in this case, see [page 394](#).

7.2.3 Automatic restart after SETS in the monitor system

The automatic restart system function in the monitor system makes it possible to automatically reload the monitor system after a system crash even under VM2000. To ensure this happens, all the preparations required for an automatic restart in BS2000 operation without VM2000 must be made (see the “Introductory Guide to System Support” [2]):

- The restart option must be activated with the BS2000 command `/SET-RESTART-OPTIONS`.
- The monitor system must have been prepared for automatic restart. This presupposes that the prerequisites for a fast or fast startup of the monitor system are satisfied, i.e. the standard file names for the required files must be set. There must be no incorrect REPs or parameters.
- On restart the main console must be the IPL console.
 - If the original IPL console is replaced as main console by a redundant SKP console during ongoing operation, VM2000 uses the new main console as the IPL console for automatic restart.
- In a system memory dump with SLED, one of the following two files must be available:
 - `$TSOS.SLEDFILE` (file size < 32 Gbytes; for SLED without parameter file)
 - `$TSOS.SYSPAR.SLED.<version>` (for SLED with parameter file)

If the `$TSOS.SLEDFILE` file already contains a memory dump (logically not empty) in the case of SLED without a parameter file, no new memory dump can be generated and the system will immediately be reloaded in accordance with the specifications from the `/SET-RESTART-OPTIONS` command.

During the SLED run and the startup of the monitor system all other guest systems are operational and can be operated using BS2000 consoles, for example. VM2000 and the virtual machines cannot, however, be addressed using VM2000 commands.

Because \$VMCONS is not available during this time, guest systems cannot be operated in the VC dialog. Guest systems which are **only** operated via VC dialog cannot be used during this time, see [page 119](#).

After SYSTEM READY of the monitor system, VM2000 can be operated again using the BS2000 consoles. After VMCONS READY has been issued, each VM administrator can reopen the dialog with his/her VM.




The monitor VM is set up again with the most recently assigned attributes and resources (exception: implicitly assigned devices are removed). The other VMs remain set up as before.

When the monitor system is restarted, the VM2000 settings of the startup parameter service are not evaluated. The restart takes place with the most recent VM2000 settings. The values in the parameter file of the VM2-MON subsystem are used for the \$VMCONS parameters, see [section “Modifying the VMCONS parameters” on page 167](#).

The guest systems are informed in the case of an automatic restart. Message NRTV002 (“Monitor system not ready. Restart has been initiated”) is output on the guest system BS2000 console. When the monitor system reaches SYSTEM READY again, message NRTV003 (“Monitor system ready again”) is displayed.

7.2.4 Restarting the monitor system via SVP

 On SQ servers the monitor system can be restarted via the SQ manager or using the SVP functions of the KVP console which is assigned to the monitor system (see the manual “Operation and Administration” [18]).

Use of the restart facility under SVP (on S servers) is dependent on the same conditions as an automatic restart of the monitor system. Exception: the restart option does not need to be set.

If the following problems occur in the monitor system, and /SHUTDOWN can no longer be entered, the monitor system can be restarted using SVP:

- problems that do not result in abnormal system termination (“hung” system in the monitor system, UCON BUSY, no input possible)
- SETS without the Restart option (VMS0018)

Initialize restart

To restart the monitor system using SVP, SVP must be called, and in the MODE SELECTION FRAME screen the RESTART function must be activated.

Restart procedure

The restart of the monitor system then proceeds in the same way as an automatic restart in BS2000/OSD (with SLED).



\$VMCONS is not available while the monitor system is being restarted automatically. Guest systems that are **only** operated via a virtual console cannot be operated during this period.

The monitor VM is set up again with the most recently assigned attributes and resources (exception: implicitly assigned devices are removed). The other VMs remain set up as before.

When the monitor system is restarted, the VM2000 settings of the startup parameter service are not evaluated. The restart takes place with the most recent VM2000 settings. The values in the parameter file of the VM2-MON subsystem are used for the \$VMCONS parameters, see [section “Modifying the VMCONS parameters” on page 167](#).

The guest systems are informed at restart via SVP. Message NRTV002 (“Monitor system not ready. Restart has been initiated”) is output on the guest system BS2000 console. When the monitor system reaches SYSTEM READY again, message NRTV003 (“Monitor system ready again”) is displayed.

7.2.5 Restarting a task after an error in a system task of VM2000

After an error in the VM2000 monitor, the system task involved (VM2M, VM2G, VM2C, VM2J, VM2P, VM2E) is continued at a consistency point.

A task restart after an error in the VM2000 monitor may have the following consequences:

- Error in the main task (VM2M)
Inconsistent table entries may exist. A VM2000 command may not have been executed completely. Information on the current VM state is supplied by the VM2000 information commands.
- Error in the general task (VM2G, S servers)
Requests to the VM2000 monitor may not have been processed completely.
- Error in the \$VMCONS task (VM2C)
All connections to \$VMCONS (ADMIN dialog, VC dialog) are cleared. Once \$VMCONS has restarted automatically, these connections must be established again.
- Error in the job variable task (VM2J)
The current update of a MONJV to monitor a VM or main memory reconfiguration is not carried out. When the monitoring is finished, e.g. with /DELETE-VM, problems may occur in resetting the MONJV. A MONJV of this kind can only be used again when the MONJV protection has been reset manually with /MODIFY-JV-ATTRIBUTES JV-NAME=... , PROTECTION=(MONJV-PROTECTION=*NO) or in the next system run of the monitor system.
The updating of further MONJVs is delayed until the VM2J task is automatically restarted.
- Error in the pubset task (VM2P)
The last action for a pubset is rejected with the message VMS4252 ("timeout").
- Error in the task for the VM2000 Agent (VM2E, SQ servers)
Requests of the VM2000 Agent may possibly not have been executed.

7.3 VMDUMP utility routine

A special **VM2000 dump** containing the relevant VM2000 data in convenient list form can be generated for the diagnosis of VM2000 errors. The VMDUMP utility routine is used to generate a VM2000 dump. The VM2000 dump is not required for regular operation of VM2000.

VMDUMP generates a VM2000 dump of the data of the current overall VM2000 system.

- S On S servers the VM2000 dump contains the data of the VM2000 monitor as well as the data of the VM2000 hypervisor.
- SQ On SQ servers the VM2000 dump contains only the data of the VM2000 monitor. It does not contain data of the Xen hypervisor or domain 0 (X2000).

Execution of VMDUMP as a separate program

To call VMDUMP as a separate program, the following operations are necessary:

1. The VMDUMP program is called under TSOS with /START-VMDUMP.

VMDUMP can also be called with /START-EXECUTABLE-PROGRAM FROM-FILE=\$<userid>.SYSPRG.VM2000-UTIL.<ver> under TSOS if the user ID <userid> under which the VM2000 dump was installed is known.

2. The input for VMDUMP are requested by the following message on the screen:
ENTER VM2000 DUMP-COMMAND (H=HELP, A=APPLICATION HINTS, E=END)
3. The permissible VMDUMP input and its functions can be listed on the screen by entering H[ELP].

```

VM2000 D U M P   -   C O M M A N D S
-----
A                OUTPUT APPLICATION INSTRUCTIONS FOR DUMP
B                PRINT DESCRIPTION OF DUMP OUTPUT
C <TEXT>        INSERT COMMENTARY LINE IN SYSLST
D                PROCESS DUMP DATA OF CURRENT SESSION
E / END         TERMINATE THE PROGRAM
H / HELP        REQUEST INFORMATION ON VM2000 DUMP COMMANDS
S                REQUEST STATUS INFORMATION ON VM2000 TRACE
```

4. The desired command can be input after the following output is displayed: ENTER VM2000-DUMP-COMMAND . . .

5. The A command supplies notes on the use of VMDUMP.

VM2000 D U M P I N S T R U C T I O N S

- 1) ERROR REPETITION (TEST PRIVILEGE (8,8))
 - DETERMINE VM2000 TRACE STATUS USING COMMAND "S"
 - SET DESIRED TRACE USING /MODIFY-VM-ATTRIBUTES
 - REPEAT ERROR
 - PRODUCE DUMP (SYSLST) USING COMMAND "D"
 - WITH INTERNAL TESTING: OUTPUT DUMP DESCRIPTION (IN GERMAN ONLY) USING COMMAND "B"
 - PRINT DUMP USING "/PRINT *SYSLST" AND SEND IT TO CUSTOMER SERVICE DEPARTMENT TOGETHER WITH CONSOLE LOG

- 2) DUMP OF RUNNING SYSTEM (TEST PRIVILEGE (8,1))
 - COMMAND "D" PRODUCES DUMP; PRINT *SYSLST

6. The E[ND] statement terminates the VMDUMP program.

Automatic VMDUMP

In the event of errors, VM2000 **automatically** calls the VMDUMP utility to generate a VM2000 dump. The file with the VM2000 dump is given the name `$SYSDUMP.SYS.VMDUMP.LST.VM<n>.<date>.<time>`, where `<n>` is the index of the VM causing the error, or 0 if it is a VM2000 hypervisor error (on S servers).



To permit VMDUMP to alter its test privilege without the consent of the operator when an automatic VM2000 dump is created, the user attribute `TEST-OPTIONS, MODIFICATION` should have the value `*UNCONTROLLED` for the TSOS user ID. This is also the default value for the TSOS user ID. See the `MODIFY-` and `SHOW-USER-ATTRIBUTES` commands in the "Commands" manual [12].

The automatically generated VM2000 dump is an important component of the diagnostic documents.

You can view the VMDUMP file with the `/SHOW-FILE` command or the editor, or you can print it out on a standard printer using the following command:

```
/PRINT-DOCUMENT <filename>, LINE-SPACING=*BY-EBCDIC-CONTROL
```

The automatic VMDUMP is generated in the following cases:

- Errors in the VM2000 monitor (message: VMS2014)
- errors in \$VMCONS



On S servers the automatic VMDUMP is also created in the case of:

- Undefined VM2000 errors in the VM2000 hypervisor (without a message)
- Errors in the hardware (message: VMS0017)

On S servers, the diagnostic data of the VM2000 hypervisor is also contained in the SLED of a guest system (dump file section VM2HYVVS).

7.4 Other diagnostic aids

In addition to VMDUMP, VM2000 also offers internal trace functions.

The following BS2000 diagnostic aids are also available for use in the monitor system:

- CONSLOG file (also contains the VM2000 commands and messages)
- REPMLOG file (also contains the VM2000 Rep corrections)

SQ On SQ servers diagnostic information can also be created via the SQ manager (menu *Administration > Diagnostics*).

Trace functions of VM2000

VM2000 offers the following internal trace functions in the monitor system:

- the execution trace for runtime logging
- the error trace for error logging
- \$VMCONS-TRACE for runtime logging in \$VMCONS

The trace functions are called by VM2000 at specific execution points to log a run or a detected error.

VM2000 writes trace entries into separate buffers, the execution trace buffer, the error trace buffer and the \$VMCONS trace buffer. The contents of these buffers are output by means of the utility routine VMDUMP.

Control of the trace functions

`TRACE-LEVEL=*OFF` is set after VM2000 READY. The scope of TRACE logging can be defined with `/MODIFY-VM-ATTRIBUTES TRACE-LEVEL=*OFF/*STD/*MAXIMUM`.

When `*OFF` is specified, only system errors such as inconsistent table statuses or undefined return codes after VM2000 hypervisor calls are logged in the execution and error trace buffers.

When `*STD` is specified, important execution points are also logged.

Specifying `*MAX` causes even more execution points to be logged.

7.5 Diagnostic file SLEDFILE

A diagnostic file (SLEDFILE) is created using the memory dump routine SLED. It can be analyzed with the DAMP diagnostic program. A detailed description of SLED and DAMP can be found in the “Diagnostics Handbook” [4].

7.5.1 Creating SLEDFILES during VM2000 operation

A SLEDFILE can be created for a guest system, for the monitor system or for the whole VM2000 system.

SLED for a BS2000 guest system that is not the monitor system

The SLED of the guest system can be created with `/START-VM . . . ,DIAGNOSTIC-IPL=*YES` or by an automatic restart after SETS in the guest system.

SQ On SQ servers the SLED can also be created using the SVP functions of the KVP console which is assigned to the VM.

See [page 290](#) for details on how to handle implicitly assigned devices in the case of `/START-VM`.

S On S servers, the diagnostic data of the VM2000 hypervisor is also contained in the SLEDFILE of a guest system (dump file section `VM2HYPVS`).

SQ On SQ servers, the SLEDFILE contains no data of the Xen hypervisor or of domain 0 (X2000).

SLED for the monitor system

The SLEDFILE also contains the diagnostic data of the VM2000 monitor.

S The SLED of the monitor system can be created when the monitor system is restarted:

- [Automatic restart after SETS in the monitor system](#) (see [page 402](#))
- [Restarting the monitor system via SVP](#) (see [page 403](#))

On S servers, the diagnostic data of the VM2000 hypervisor is also contained in the SLED of a guest system (dump file section VM2HYPVS).

SQ The SLED of the monitor system can be created by:

- An automatic restart after SETS in the monitor system (see [page 402](#))
- Loading the SLED explicitly via the SKP functions of the KVP console which is assigned to the monitor VM

The SLEDFILE contains no data of the Xen hypervisor or of domain 0 (X2000).

SLED for the whole VM2000 system (S servers)

The SLED of the entire system can be created by explicitly loading the SLED in the same way as it is loaded in BS2000 operation without VM2000 or at a restart after VM2000 hypervisor termination (see [page 401](#)).

With the non-standard SLED you can specify the extent of the SLED output using the MODE parameter in response to message NSD3001:

- MODE=REAL
The entire main memory is output.
- MODE=STD
The VM2000 hypervisor main memory and the entire monitor system are output.
- MODE=ALL
The entire main memory and the complete monitor system are output.

Normally, SLED is output with MODE=STD (response Y or EOT to message NSD3001).

MODE=REAL or MODE=ALL should only be specified if it is clear that the main memory of the other VMs is required to diagnose the error in the VM2000 hypervisor or the monitor VM. This is only the case in exceptional situations.

7.5.2 Analyzing SLEDFILES with DAMP

The DAMP diagnostic routine enables a SLEDFILE to be diagnosed, see the “Diagnostics Handbook“ [4]. As the data of the VM2000 monitor is also contained in the SLED of a monitor system, VM2000 problems can also be analyzed here.

Analyzing the SLEDFILE for the monitor system or another guest system

On S servers, the diagnostic data of the VM2000 hypervisor is also contained in the SLED of a guest system (dump file section VM2HYPVS).

The area of the dump file section VM2HYPVS can be addressed and output in a dump window (W4 to W9, W21-W99) by DAMP:

ASEL='SCT' The ASEL input field (address space selector) determines the output of a dump file section in this case.

ASID='VM2HYPVS'

The ASID input field (address space identifier) contains the name of the dump file section.

```
DAMP V4.7A00 SLED(18.0) from BS2000(18.0)      <date> <time>

                                +00000= 001000 SCT=VM2HYPVS          W4,D ,L19
00001000 (00000):D7C9C401 00001024 00000000 E5D4F2F0 <==> PIDVM20
00001010 (00010):F0F04040 E5F1F04B F0C1F0F0 F0F00000 <==> 00 V10.0A0000
00001020 (00020):00000000 00470101 FFFFFFFF D7C9C401 <==> 8888PID
00001030 (00030):00000000 00000000 E2E8E2E2 E3C1D9E3 <==> SYSSTART
00001040 (00040):E5F1F84B F0C1F0F0 D6F10601 00000000 <==> V18.0A0001
00001050 (00050):00000000 00000010 E85CE3C5 E2E3C1D4 <==> Y*TESTAM
00001060 (00060):C5D5E35C 00000000 01000000 00E8C4D6 <==> ENT*YDO
```

Figure 31: Output as of address 1000 in the dump file section VM2HYPVS

Analyzing the SLEDFILE for the whole VM2000 system (S servers)

A SLED with a VM2000 dump object (SLED of the whole VM2000 system) or a SLED with a virtual machine dump object can be analyzed using DAMP.

In the input field “mode select”, columns 74-76 for the status window (W2) of DAMP, the information output can be controlled by various modes (INF, TSK, PLK, SLK). The INF mode is set automatically if the dump file contains more than one object, e.g. in the case of a SLED with the dump object VM2000.

In INF mode, the following information on the dump object VM2000 is displayed:

- type of the dump generator of the object to be analyzed,
- contents of the product ID; these include the name and version of the product, as well as the address of the so-called dump testament, if any (contains internal SLED information).
- overview of the virtual machines that have been created in the overall system.
- overview of the virtual machines contained in the dump. By default, the VM2000 hypervisor (HYP) is selected with absolute addressing. A guest system on the VM (VM01, VM02, ...) is selected by marking.

```
DAMP V4.7A00 SLED(18.0) from VM2000(10.0)

Analyzed Object: VM2000   V10.0A0000           TID=           TSN=           W2,INF,L19
Dumpfile:   :4IVZ:$TSOS.SLEDFILE
                                           (No Selection)

Generator Name: SLED      (STD)           Generator Version: 180.01
Product Name:   VM2000           Product Version:   V10.0A0000
Address of Dump-Testament: 00001024 (absolut)

Time of creating SLED:  <date> <time>
Information about VM2000: Hypervisor pages FROM 0000 TO 0FFF
VMs created by VM2000:  HYP  VM01 VM02 VM03 VM04 VM05 VM06

VMs dumped by SLED and their page boundaries
VM01: 001000 - 020FFF

CMD:
Key: 1=Help 2=Inf 3=PCB 4=Dump  5=Dump  6=Dump  7=Dump  8=Dump  9=Dump
```

Figure 32: Information screen in the status window (W2). Dump generator SLED. Dump object VM2000

In the input field ASEL (address-space selector) for a dump window (W4 through W9, W21-W99) of DAMP, real or absolute addressing is possible with the dump object VM2000:

ASEL='ABS' The address space is the absolute main memory. The VM2000 guest system can be addressed with absolute addresses:

```
DAMP V4.7A00 SLED(18.0) from VM2000(10.0)      2012-01-20 13:02:02

                                +00000= 001000 ABS=00000000          W5,D ,L19
00001000 (00000):D7C9C401 00001024 00000000 E5D4F2F0 <==> PIDVM20
00001010 (00010):F0F04040 E5F1F04B F0C1F0F0 F0F00000 <==> 00 V10.0A0000
00001020 (00020):00000000 00470101 FFFFFFFF D7C9C401 <==> BBBBPID
00001030 (00030):00000000 00000000 E2E8E2E2 E3C1D9E3 <==> SYSSTART
00001040 (00040):E5F1F84B F0C1F0F0 D6F10601 00000000 <==> V18.0A0001
00001050 (00050):00000000 00000010 E85CE3C5 E2E3C1D4 <==> Y*TESTAM
00001060 (00060):C5D5E35C 00000000 01000000 00E8C4D6 <==> ENT*YDO
```

Figure 33: Output from absolute address 1000 in the dump object VM2000

ASEL='RM' The address space is the real main memory in the selected guest system:

```
DAMP V4.7A00 SLED(18.0) from VM2000(10.0)      2012-01-20 13:02:02

                                +00000= 001000 RM =00000000          W4,D ,L19
00001000 (00000):D7C9C401 00000000 00000000 C2E2F2F0 <==> PIDBS20
00001010 (00010):F0F04040 E5F1F84B F0C1F0F0 D6F10000 <==> 00 V18.0A0001
00001020 (00020):00000000 E7E5E3F4 FFFFFFFF 72C96008 <==> XVT4BBBBBI-
00001030 (00030):7DAB4040 710011B8 FFFFFFFF 71001274 <==> ' BBBB
00001040 (00040):714167C0 710011A4 7F67CEEO 72D93280 <==> u"R
00001050 (00050):71001208 710016F0 FFFFFFFF 71001208 <==> 0BBBBB
00001060 (00060):FFFFFFF FFFFFFFF FFFFFFFF 7FBDB000 <==> BBBB BBBB BBBB "Ü
```

Figure 34: Output from real address 1000 in a guest system

8 VM2000 commands

This chapter contains a description of all VM2000 commands. The command syntax is explained in the "Commands" manual [12].

The section [section "VM2000 Operating strategy" on page 88](#) describes the input options and input media for the VM2000 commands.

You can find a description of the privileged functions in the commands as of [page 190](#).

Notational conventions

The command descriptions contain a great deal of information which only applies for S server or SQ server series. These are labeled as follows:

- S This comment applies only for S servers
- SQ This comment applies only for SQ servers

8.1 Overview of commands for VM2000 operation

VM2000 commands can be split up into two groups:

- VM2000 administrator commands e.g. /CREATE-VM
- VM administrator commands e.g. /START-VM)

VM2000 administrator commands

- are normally intended for use by the VM2000 administrator (who is also the VM administrator of the monitor VM)
- can be entered in an ADMIN dialog with the monitor VM (recommendation) and via BS2000 consoles or logical consoles in the monitor system
- require the OPERATING privilege or the VM2000-ADMINISTRATION privilege in the ADMIN dialog via a user task
- are used to operate the whole VM2000 system.

VM administrator commands

- are normally intended for use by the VM administrator
- can only be input by the VM administrator in the ADMIN dialog for his/her VM
- require the OPERATING privilege or the VIRTUAL-MACHINE-ADMINISTRATION privilege in the ADMIN dialog via a user task
- are used to operate a virtual machine
- can be used by the VM2000 administrator for any other VMs
- can have operand values whose use is restricted to the VM2000 administrator

Exceptions arise when the command/function range is extended and reduced for the VM and VM2000 administrator (ADMIN-PRIVILEGES, see [page 192](#)).

8.1.1 Table of VM2000 commands

The following table lists the VM2000 commands in alphabetical order.

Command name	For VM2000 administrator	VM Administrator	Privileged operands (VM-ID=)
ADD-VM-DEVICES	x	x ¹	
ADD-VM-RESOURCES	x	x ¹	
ADD-VM-TO-VM-GROUP	x		
ASSIGN-VM-GROUP-TO-CPU-POOL	x		
ASSIGN-VM-TO-CPU-POOL	x		
ATTACH-VM-RESOURCES	x		
BEGIN-VM-DIALOG	x	x	
CALL-VM-PROCEDURE	x ²	x ²	
CREATE-VM	x		
CREATE-VM-ASSIGNMENT-SET	x		
CREATE-VM-CPU-POOL	x		
CREATE-VM-GROUP	x		
DELETE-VM	x	x	*ALL
DELETE-VM-ASSIGNMENT-SET	x		
DELETE-VM-CPU-POOL	x		
DELETE-VM-GROUP	x		
DETACH-VM-RESOURCES	x		
END-VM-DIALOG	x	x	
EXTEND-VM-MEMORY	x		
HOLD-VM	x	x	*ALL
MODIFY-VM-ATTRIBUTES	x	x ¹	
MODIFY-VM-DEVICE-ATTRIBUTES	x		
MODIFY-VM-DEVICE-USAGE	x	x ¹	
MODIFY-VM-GROUP-ATTRIBUTES	x		
MODIFY-VMCONS-PARAMETERS	x		
MOVE-VM	x		
REDUCE-VM-MEMORY	x		
REMOVE-VM-DEVICES	x	x	
REMOVE-VM-FROM-VM-GROUP	x		
REMOVE-VM-RESOURCES	x	x	

Command name	For VM2000 administrator	VM Administrator	Privileged operands (VM-ID=)
RESET-VM	x	x	
RESTORE-VM-CONFIGURATION	x		
RESUME-VM	x	x	*ALL
SAVE-VM-CONFIGURATION	x		
SEND-VM-MESSAGE	x	x	
SHOW-VM-ASSIGNMENT-SET	x		
SHOW-VM-ATTRIBUTES	x	x	*VM2000 *BY-VM-GROUP *BY-CPU-POOL <name 1..8 with-wild(20)>
SHOW-VM-CPU-POOL	x		
SHOW-VM-DEVICE-STATUS	x	x ¹	
SHOW-VM-GROUP	x		
SHOW-VM-RESOURCES	x	x ¹	
SHOW-VM-STATUS	x		
SHUTDOWN-VM	x	x	*VM-GUESTS *VM2000
START-VM	x	x	
SWITCH-VM-CPU	x		
SWITCH-VM-DEVICES	x		

¹ Command available to VM administrators with ADMIN-PRIVILEGES=*MAXIMUM (see [page 193](#))

² Command available only in ADMIN dialog via \$VMCONS

8.1.2 Grouping of VM2000 commands according to function

Operating a VM

Command	Function	Authorization	Privileged operands (VM-ID=)
CREATE-VM	Initializes a VM. Thus the VM is assigned attributes and resources.	VM2000 administrator	
MODIFY-VM-ATTRIBUTES	Modifies VM attributes	VM2000 administrator ¹	
ADD-VM-RESOURCES	Assigns global storage to a VM	VM2000 administrator ¹	
REMOVE-VM-RESOURCES	Cancels assignment of global storage to a VM.	VM administrator	
START-VM	Starts a guest system on a VM	VM administrator	
HOLD-VM	Places a VM in the wait state	VM administrator	*ALL
RESUME-VM	Cancels the wait state for a VM	VM administrator	*ALL
EXTEND-VM-MEMORY	Increases main memory for a VM.	VM2000 administrator	
REDUCE-VM-MEMORY	Reduces main memory for a VM.	VM2000 administrator	
MOVE-VM	Moves a VM in main memory	VM2000 administrator	
SHOW-VM-ATTRIBUTES	Shows the VM attributes	VM administrator	*VM2000 *BY-VM-GROUP *BY-CPU-POOL <name 1..8> with-wild(20)>
RESET-VM	Aborts the guest system on a VM	VM administrator	
SHUTDOWN-VM	Executes the shutdown in the guest system	VM administrator	*VM-GUESTS *VM2000
DELETE-VM	Terminates a VM	VM administrator	*ALL

¹ Command available to VM administrators with ADMIN-PRIVILEGES=*MAXIMUM (see [page 193](#))

Operation of VM groups (S servers)

Command	Function	Authorization
CREATE-VM-GROUP	Creates a VM group	VM2000 administrator
ADD-VM-TO-VM-GROUP	Adds a VM to a VM group	VM2000 administrator
REMOVE-VM-FROM-VM-GROUP	Removes a VM from a VM group	VM2000 administrator
MODIFY-VM-GROUP-ATTRIBUTES	Modifies the attributes of a VM group	VM2000 administrator
SHOW-VM-GROUP	Shows attributes of VM groups	VM2000 administrator
DELETE-VM-GROUP	Deletes a VM group	VM2000 administrator

Managing devices

Command	Function	Authorization
CREATE-VM-ASSIGNMENT-SET	Create assignment set	VM2000 administrator
DELETE-VM-ASSIGNMENT-SET	Deletes an assignment set	VM2000 administrator
SHOW-VM-ASSIGNMENT-SET	Shows the assignment sets created	VM2000 administrator
ADD-VM-DEVICES	Allocates available devices to a VM	VM2000 administrator ¹
SWITCH-VM-DEVICES	Deallocates devices from a VM and allocates them to another VM	VM2000 administrator
MODIFY-VM-DEVICE-ATTRIBUTES	Assigns device attributes	VM2000 administrator
MODIFY-VM-DEVICE-USAGE	Changes the usage mode of devices of a VM	VM2000 administrator ¹
SHOW-VM-DEVICE-STATUS	Shows detailed device information	VM2000 administrator ¹
REMOVE-VM-DEVICES	Deallocates devices from a VM	VM administrator

¹ Command available to VM administrators with ADMIN-PRIVILEGES=*MAXIMUM (see [page 193](#))

Managing CPU pools

Command	Function	Authorization
CREATE-VM-CPU-POOL	Creates a CPU pool	VM2000 administrator
ASSIGN-VM-GROUP-TO-CPU-POOL	Assigns a VM group to another CPU pool	VM2000 administrator
ASSIGN-VM-TO-CPU-POOL	Assigns a VM to another CPU pool	VM2000 administrator
SWITCH-VM-CPU	Deletes CPUs from a CPU pool and assigns them to another CPU pool	VM2000 administrator
SHOW-VM-CPU-POOL	Shows attributes of CPU pools	VM2000 administrator
DELETE-VM-CPU-POOL	Deletes a CPU pool	VM2000 administrator

Managing the VM2000 configuration

Command	Function	Authorization
SAVE-VM-CONFIGURATION	Saves the current VM2000 configuration	VM2000 administrator
RESTORE-VM-CONFIGURATION	Restores the saved VM2000 configuration	VM2000 administrator

VM2000 management

Command	Function	Authorization
SHOW-VM-RESOURCES	Shows the current status of VM2000 resources	VM2000 administrator ¹
SHOW-VM-STATUS	Outputs monitored values on the utilization of the overall VM2000 system	VM2000 administrator
ATTACH-VM-RESOURCES	Attach real CPUs	VM2000 administrator
DETACH-VM-RESOURCES	Detaches real CPUs	VM2000 administrator

¹ Command available to VM administrators with ADMIN-PRIVILEGES=*MAXIMUM (see [page 193](#))

Administration and operating (ADMIN dialog, VC dialog)

Command	Function	Authorization
BEGIN-VM-DIALOG	Starts administration or operation	VM administrator
END-VM-DIALOG	Terminates administration or operation	VM administrator
MODIFY-VMCONS-PARAMETERS	Modifies the parameters for \$VMCONS	VM2000 administrator
CALL-VM-PROCEDURE ¹	Starts a VM2000 procedure file with VM2000 commands	VM administrator
SEND-VM-MESSAGE	Sends a message to VM administrators	VM administrator

¹ Command available only in ADMIN dialog via \$VMCONS

8.2 Command return codes

VM2000 returns information on the analysis of the command entered and on the execution of the command in a command return code. This command return code can be compared with a return code in a program and allows you to react appropriately to specific errors.

The command return code comprises three parts:

- the maincode, a message code that can be specified with the `HELP-MSG-INFORMATION` command in order to obtain detailed information.
- subcode 1, which assigns the error situation to an error class that indicates how serious an error is.
- subcode 2, which can contain supplementary information (a value other than zero).

Command return codes which can be returned by a VM2000 command on execution form part of the command description. In each case, a table follows the command description showing subcode 1, subcode 2, the main code and the meaning. VM2000 knows the following general return codes:

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
	1	CMD0202	Syntax or semantic error
	32	CMD0221	Internal system error
	64	CMD0216	Required privilege not present
	65	CMD2241	Subsystem not loaded
	128	CMD2280	Subsystem not available
	128	VMS4504	Memory bottleneck
	128	VMS4513	VM2000 Hypervisor bottleneck
	130	VMS6000	VM2000 Agent not available
	130	VMS6001	Administration component of the carrier system not available
	64	VMS9001	Function not available for input medium or guest system
	64	VMS9004	Functionality not supported on current HSI or with current parameter assignment



Even when SC1 is not equal to zero and with the setting `ERROR-MECHANISM=*SPIN-OFF-COMPATIBLE (SET-/MODIFY-PROCEDURE-OPTIONS)`, VM2000 information commands do **not** trigger the spin-off mechanism.

Command return codes in a user task

When VM2000 commands are input via a privileged user task, the command return code is stored in standardized S variables and can be analyzed with the built-in functions of `SDF-P-BASYS - MAINCODE`, `SUBCODE1` and `SUBCODE2`, see the “Commands“ manual [12].

Command return codes via \$VMCONS or at a BS2000 console

With command input via \$VMCONS or from a BS2000 console, the command name and the command return code are output as inserts of the standard termination message

NBR0740 (guaranteed message), e.g.

```
NBR0740 COMMAND COMPLETED 'ADD-VM-DEVICES';
      (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
```

Commands, operands and operand values which are not supported

Some of the VM2000 commands, operands and operand values have no meaning on certain server series and are not supported on them, see also “[Notational conventions](#)” on [page 415](#). If they are entered despite this, they are rejected by VM2000 with a message and a return code.

Entry	Message	Return code, see page 423
Command	VMS3011	0, 64, VMS9004
Operand	VMS3012 VMS3030	0, 64, VMS9004
Operand value	VMS3032	0, 64, VMS9004

8.3 Guaranteed messages

For a number of VM2000 messages, message codes and inserts (number and semantics) are guaranteed as fixed components for future VM2000 versions. Such messages are called guaranteed messages.

Guaranteed messages of VM2000 are listed under the relevant commands in the “Guaranteed messages” section.

If the software product SDF-P is used, VM2000 outputs the guaranteed messages in structured S variables. Via S variables you can access certain message data directly without having to know the output layout of the messages. It is thus possible to control further processing in S procedures depending on the contents of these variables.

Guaranteed messages of VM2000 are listed under the relevant commands in the “Guaranteed messages” section.

For information on working with S variables, see the “Commands” manual [[12](#)].

8.4 SHOW output in S variables

VM2000 shows output in S variables for the information commands of VM2000 (with the exception of /SHOW-VM-STATUS) if these commands are executed in a user task. The S variables can be analyzed with the software product SDF-P.

The output in S variables is a component of the respective command descriptions. The S variables are listed in table form at the end of the command description, with their name, meaning and content.

In a structured output in S variables the current individual items of information are each stored as the contents of an S variable. Via the names of the S variables the user accesses the desired item of information in the information command, regardless of the output layout. The names of the S variables are preset for each information command and guaranteed for the follow-up versions, i.e. they will not be changed.

Name components of the S variables

The names of the list elements (e.g. VM-INDEX) are preset for the various information commands. They are appended to the S variable names declared by the user. The names are oriented as far as possible toward the corresponding operand names of output designations. If the information at this level continues to be hierarchically subdivided, further names are appended, each separated by a period.

Name components of the S variables

As far as possible, the contents of the S variables correspond to the associated operand values. The data type of the S variable determines the character set from which the contents of the S variable come.

Contents of the S variables

In the “Output in S variables” sections, the tables are sorted according to application groups. The S variables have a defined data type: string, integer or Boolean. The column T (type) indicates the data type of the contents: S (string), I (integer), B (Boolean).

For information on working with S variables, see the “Commands“ manual [12].

For an example of the evaluation of S variables under VM2000, turn to [page 133](#).

8.5 Description of the commands

The commands are described alphabetically under the following headings:

- command name and function
- description of command function
- command name and function
- description of command function
- representation of command format
- description of command operands
- description of the command return codes
- list of guaranteed messages
- for information commands: output in S variables

Command name	Function
ADD-VM-DEVICES	Allocate devices to a VM
ADD-VM-RESOURCES	Assign global storage to a VM
ADD-VM-TO-VM-GROUP	Add a VM to a VM group
ASSIGN-VM-GROUP-TO-CPU-POOL	Assign a VM group to a CPU pool
ASSIGN-VM-TO-CPU-POOL	Assign a VM to a CPU pool
ATTACH-VM-RESOURCES	Attach real CPUs
BEGIN-VM-DIALOG	Start administration or operation
CALL-VM-PROCEDURE	Start a VM2000 procedure file
CREATE-VM	Initialize a VM
CREATE-VM-ASSIGNMENT-SET	Create assignment set
CREATE-VM-CPU-POOL	Create a CPU pool
CREATE-VM-GROUP	Create a VM group
DELETE-VM	Terminate a VM
DELETE-VM-ASSIGNMENT-SET	Deletes an assignment set
DELETE-VM-CPU-POOL	Delete CPU pool
DELETE-VM-GROUP	Delete a VM group
DETACH-VM-RESOURCES	Detach real CPUs
END-VM-DIALOG	Terminate administration or operation

Command name	Function
EXTEND-VM-MEMORY	Increase main memory for a VM
HOLD-VM	Suspend VM
MODIFY-VM-ATTRIBUTES	Modify attributes of a VM
MODIFY-VM-DEVICE-ATTRIBUTES	Modify device attributes
MODIFY-VM-DEVICE-USAGE	Modify usage mode of the devices of a VM
MODIFY-VM-GROUP-ATTRIBUTES	Modify attributes of a VM group
MODIFY-VMCONS-PARAMETERS	Modify parameters of \$VMCONS
MOVE-VM	Move a VM in main memory
REDUCE-VM-MEMORY	Reduce main memory for a VM
REMOVE-VM-DEVICES	Deallocate devices from a VM
REMOVE-VM-FROM-VM-GROUP	Remove a VM from a VM group
REMOVE-VM-RESOURCES	Cancel the assignment of global storage to a VM
RESET-VM	Aborts the guest system on a VM
RESTORE-VM-CONFIGURATION	Restores the saved VM2000 configuration
RESUME-VM	Resume VM operation
SAVE-VM-CONFIGURATION	Saves the current VM2000 configuration
SEND-VM-MESSAGE	Pass messages to VM administrators
SHOW-VM-ASSIGNMENT-SET	Output assignment sets
SHOW-VM-ATTRIBUTES	Output attributes of a VM
SHOW-VM-CPU-POOL	Output attributes of a CPU pool
SHOW-VM-DEVICE-ATTRIBUTES	Output detailed device information
SHOW-VM-GROUP	Output attributes of a VM group
SHOW-VM-RESOURCES	Output the status of resources of VM2000
SHOW-VM-STATUS	Output VM2000 monitored values
SHUTDOWN-VM	Execute a shutdown in the guest system and for VM2000
START-VM	Start a guest system on a VM
SWITCH-VM-CPU	Delete CPUs from a CPU pool and add them to another CPU pool
SWITCH-VM-DEVICES	Deallocate devices from one VM, allocate them to another

ADD-VM-DEVICES

Allocate devices to a VM

`/ADD-VM-DEVICES` is used to assign individual devices or all devices of a pubset to a VM explicitly. Individual devices are addressed using their mnemonic device names. The devices of a pubset are determined by means of the catalog ID of the pubset or the mnemonic device name of the pubres device.

This command is available only to the VM2000 administrator and VM administrators with `ADMIN-PRIVILEGES=*MAXIMUM` and can only be entered after the VM has been initialized using `/CREATE-VM`.



The devices must then be attached in the guest system so that they can be addressed in this system. A pubset must be imported in the guest system before it is used.



Alias devices (PAV) cannot be assigned explicitly.



On SQ servers no devices can be assigned explicitly to a VM while it is being migrated (Live Migration).

ADD-VM-DEVICES

```

UNITS = *PUBSET-DEVICES(...) /
          list-poss(28): *DEVICE-RANGE(...) / <alphanum-name 1..4 with-wild(20)>
*PUBSET-DEVICES(...)
  |
  | PUBSET = *BY-PUBRES-DEVICE(...) / <cat-id 1..4>
  |   *BY-PUBRES-DEVICE(...)
  |     |
  |     | UNIT = <alphanum-name 2..2> / <x-text 4..4>
  |
  | *DEVICE-RANGE(...)
  |   |
  |   | FROM = <alphanum-name 2..2> / <x-text 4..4>
  |   | ,TO = <alphanum-name 2..2> / <x-text 4..4>
  |
  | ,VM-IDENTIFICATION = <integer 1..99> / <name 1..8>
  | ,TYPE = *NONE / *SD / *VC

```

UNITS =

Specifies the devices to be assigned to a VM.

UNITS = *PUBSET-DEVICES(...)

All devices of a pubset are to be assigned to the VM.



See also [“Notes on pubset specification in the VM2000 commands” on page 313](#) and [“Pubset reconfiguration” on page 315](#).

PUBSET = *BY-PUBRES-DEVICE(...)

Determines the devices of the pubset by means of the pubres device.

UNIT = <alphanum-name 2..2> / <x-text 4..4>

Mnemonic device name of the pubres device.

PUBSET = <cat-id 1..4>

Catalog ID of the pubset. The pubset must be entered together with its pubres device in the monitor system’s MRS catalog.

UNITS = *DEVICE-RANGE(...)

All the devices in this range of names are to be assigned to the VM.



Device names `<x-text 4..4>` (in `*DEVICE-RANGE`) are selected from FROM to TO in hexadecimal order. Otherwise, device names `<alphanum-name 2..2>` are selected in alphabetical order.

If UNITS is specified with `*DEVICE-RANGE` or with wildcards, alias devices are ignored.

FROM = <alphanum-name 2..2> / <x-text 4..4>

Defines the start of the name range.

TO = <alphanum-name 2..2> / <x-text 4..4>

Defines the end of the name range. The mnemonic device name must be in the same value range (2- or 4-digit mn) as the FROM device name and must be greater than or equal to the FROM device name in ascending order in the value range.

UNITS = <alphanum-name 1..4 with-wild(20)>

Defines the devices to be assigned to the VM.

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>

Identifies the virtual machine by means of a VM index or VM name.

TYPE =

Defines the devices to be assigned to the VM.

TYPE = *NONE

The device is assigned exclusively (usage mode `EXCLUSIVE`). Exclusively assigned devices are listed in the information commands after the keyword `EXCL`.

TYPE = *SD

The device is declared as shareable (shared device, usage mode `SHARED`). The device can be assigned to several VMs at the same time (see also [page 338](#)). Shared devices are listed in the information commands after the keyword `SHARED`.

TYPE = *VC

The device is assigned exclusively as a virtual console (S servers). It must be appropriately generated (see [page 45](#)). Only one device for each VM can be operated as a virtual console. A virtual console cannot be assigned to the monitor VM. It is not permissible to specify a list in the `UNITS` operand for `TYPE=*VC`. Virtual consoles are marked in the information commands of VM2000 with the suffix `(VC)`.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
1	0	CMD0001	Device or pubset already assigned, no action required
1	0	VMS4035	No corresponding device found (perhaps only alias devices were specified)
1	0	VMS4180	Specified set of devices empty
2	0	CMD0001	Device already implicitly assigned by the guest system, device now explicitly assigned
2	0	VMS4221	Not all of the pubset's devices could be assigned
2	0	VMS4222	Only the pubres device could be assigned
	64	VMS3002	Invalid operand value (e.g. *DEVICE=RANGE operand)
	64	VMS3003	Invalid combination of operands
	64	VMS4000	VM not yet created
	64	VMS4004	Device already assigned to a VM
	64	VMS4006	Device not available
	64	VMS4016	Device cannot be used as a virtual console
	64	VMS4026	VM already has a virtual console
	64	VMS4032	Device cannot be assigned to several VMs at the same time
	64	VMS4033	Maximum number of sharers already reached
	64	VMS4034	Device is currently in shared use
	64	VMS4060	Only one virtual console possible per VM
	64	VMS4064	No channel path available
	64	VMS4067	Resource bottleneck in the hardware
	64	VMS4089	Virtual console must be assigned with TYPE=*VC
	64	VMS4170	I/O execution mode cannot be set to "indirect"
	64	VMS4190	Device is an alias device (PAV)
	64	VMS4223	Command not executed (operand *PUBSET-DEVICES)
	64	VMS9005	Command not or only partially processed
	128	VMS3042	VM being migrated. Command rejected

Guaranteed messages

VMS4004, VMS4005, VMS4016, VMS4026, VMS4030, VMS4032, VMS4033, VMS4034, VMS4064, VMS4067, VMS4068, VMS4088, VMS4089, VMS4170

Only if *PUBSET-DEVICES is specified: VMS4220, VMS4221, VMS4222, VMS4223, VMS4234

Only if UNITS is specified without *DEVICE-RANGE and without wildcards: VMS4006, VMS4190

Example

```

/add-vm-devices unit=*pubset-devices(pubset=4ivw),vm-id=system2 _____ (1)
VMS4005 DEVICE 'E059' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2) _____ (2)
VMS4005 DEVICE 'E05A' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
VMS4005 DEVICE 'E05B' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
VMS4220 PUBSET '4IVW' WITH PUBRES DEVICE 'E059' ASSIGNED TO VM (2,SYSTEM2)

/add-vm-devices (ch,ci),vm-id=system2 _____ (3)
VMS4005 DEVICE 'CH' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
VMS4005 DEVICE 'CI' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)

/add-vm-devices e062,vm-id=system2,type=*sd _____ (4)
VMS4005 DEVICE 'E062' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2) (TYPE=SD)

/add-vm-devices cc,vm-id=system2,type=*vc _____ (5)
VMS4005 DEVICE 'CC' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2) (TYPE=VC)

/show-vm-attributes information=*units,vm-id=system2
VM-ID          UNITS
  2 SYSTEM2    ASSIGNED BY VM-ADMIN _____ (6)
                EXCL: CC(VC),CH,CI,E059,E05A,E05B
                SHARED: E062


```

- (1) The VM2000 administrator assigns the home pubset devices to VM2.
- (2) VM2000 reports that the devices have been successfully assigned. Pubset 4IVW is entered together with its pubres device E059 in the monitor system's MRS catalog and is therefore already detected here by VM2000.
- (3) The VM2000 administrator assigns the KVP device pair to VM2.
- (4) A disk is assigned for shared use with other VMs.
- (5) The VM2000 administrator assigns the KVP console device pair to VM2.
- (6) List of the mnemonic device names that are explicitly assigned to the VM2. EXCL: indicates the devices that are assigned for exclusive use. The (VC) suffix designates a virtual console. The list for SHARED: contains the devices that are assigned for shared use with other VMs.

See also the detailed examples for device and pubset management on [page 294](#) and on [page 324](#).

ADD-VM-RESOURCES

Assign global storage to VM (S servers)

 Global storage is not available on SQ servers. The command is rejected with message VMS4110 and with general return code VMS9004.

`/ADD-VM-RESOURCES` is used to make a real assignment of the entire global storage to a VM or a virtual assignment of a partition of global storage to a VM.

This command is available only to the VM2000 administrator and VM administrators with `ADMIN-PRIVILEGES=*MAXIMUM` and can only be entered after the VM has been initialized using `/CREATE-VM`.

ADD-VM-RESOURCES
<pre> GS = *REAL / *VIRTUAL(...) *VIRTUAL(...) PARTITION-ID = VIRTGS01 / VIRTGS02 / ... / VIRTGS15 / VIRTGS16 ,VM-IDENTIFICATION = <integer 1..99> / <name 1..8> ,VM-SHARED = *NO / *YES </pre>

GS =

Defines the part of global storage to be assigned. Only one partition (virtual) or the entire global storage (real) can be assigned to each VM.

GS = *REAL

The entire global storage is to be assigned to the VM (real).

GS = *VIRTUAL(PARTITION-ID = VIRTGS01 / VIRTGS02 / ... / VIRTGS15 / VIRTGS16)

Specifies the partition of global storage to be assigned to the VM (virtual assignment).



No virtual global storage can be assigned to the monitor VM.

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>

Identifies the virtual machine by means of a VM index or VM name.

VM-SHARED = *NO / *YES

Defines whether global storage is to be used exclusively by one VM (*NO) or whether it can be shared by several VMs (*YES).

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	64	VMS4000	VM not yet created
	64	VMS4112	GS partition not available
	64	VMS4113	Global storage is already assigned to virtual machine
	64	VMS4117	Global storage already assigned exclusively to another VM
	64	VMS4118	Global storage is not exclusively available
	64	VMS4121	Global storage is not exclusively available
	64	VMS4122	Connection of global storage failed
	64	VMS4123	GS partition cannot be assigned in dual mode
	64	VMS9004	Functionality not supported on current HSI or with current parameter assignment

Example

Prerequisites:

- The entire real global storage has already been assigned to the monitor VM in the configuration file (see [page 58](#)). It is to be virtualized and distributed across the VMs.
- The partitions VIRTGS01, VIRTGS03 and VIRTGS04 have been set up in the monitor system.

```

/show-gs-status select=*all _____ (1)
SUBSYSTEM PARAMETER:
GS-USAGE = GLOBAL

GS-UNITS:
GS-UNIT   BEGIN      END      SIZE      FREE  CONTIGUOUS  STATE
  1         0          8191    8192     7675      7675  ATTACHED
  2       131072    139263  8192     7803      7803  ATTACHED

PARTITIONS:
PART-ID  ACCESS DATA  MODE  GSU      SIZE      BEGIN      END  ATT-DUAL  HOST/SYSID
VIRTGS04 NONE   INVAL  MONO  1        128        2        129
VIRTGS03 NONE   INVAL  DUAL  1/2      128       7805     7932  ALLOW
VIRTGS01 NONE   INVAL  DUAL  1/2      256       7933     8188  ALLOW
NSM      SHARED  VALID  DUAL  1/2      3         8189     8191  ALLOW    D017ZE14/85

FREE MONO GS-SPACE:
GS-UNIT   SIZE      BEGIN      END
  1        7675      130        7804
  2        7803     131074    138876

FREE DUAL GS-SPACE:
GS-UNIT   SIZE      BEGIN      END
  1/2      7675      130        7804

```

```

/show-vm-resources information=*gs _____ (2)
VM-ID      PARTITION-ID ATTR  GS-SIZE  MODE  STATUS
1 VM0001   *REAL      SH      16384 MB DUAL  ASS
2 VM0002   NO GS     ASSIGNED
3 VM0003   NO GS     ASSIGNED
4 VM0004   NO GS     ASSIGNED
/add-vm-resources gs=*virtual(partition-id=virtgs03),vm-id=2 _____ (3)
/add-vm-resources gs=*virtual(partition-id=virtgs01),
                  vm-id=3,vm-shared=*yes
/add-vm-resources gs=*virtual(partition-id=virtgs01),
                  vm-id=4,vm-shared=*yes _____ (4)
/show-vm-resources information=*gs _____ (5)
VM-ID      PARTITION-ID ATTR  GS-SIZE  MODE  STATUS
1 VM0001   *REAL      SH      16384 MB DUAL  ASS
2 VM0002   VIRTGS03             256 MB DUAL  ASS
3 VM0003   VIRTGS01      SH      512 MB DUAL  ASS
4 VM0004   VIRTGS01      SH      512 MB DUAL  ASS

```

- (1) With this BS2000 command, the VM2000 administrator fetches information on the assignment of the global storage in the monitor system.
- (2) The VM2000 administrator fetches information on the distribution of global storage in VM2000.
- (3) The partition VIRTGS03 is to be assigned exclusively to VM2.
- (4) The partition VIRTGS01 is to be assigned to VM3 and VM4 for shared use.
- (5) Assignment of the virtual global storage partitions to the VMs was successful.

ADD-VM-TO-VM-GROUP

Add a VM to a VM group (S servers)

SQ No VM groups are available on SQ servers. The command is rejected with message VMS3011 and with general return code VMS9004.

/ADD-VM-TO-VM-GROUP is used to add a VM to a VM group.

This command is only permitted for the VM2000 administrator.

The VM may not already have been added to a VM group.

The VM must be assigned to the CPU pool of the VM group.

ADD-VM-TO-VM-GROUP
VM-IDENTIFICATION = <integer 1..99> / <name 1..8> VM-GROUP-NAME = <name 1..8> MEMBER-CPU-QUOTA = <fixed 0.01..99.99>

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>

Identifies the virtual machine by means of a VM index or VM name.

VM-GROUP-NAME = <name 1..8>

Name of the VM group.

MEMBER-CPU-QUOTA = <fixed 0.01..99.99>

Member CPU quota of the VM.

Determines the VM's own CPU share in comparison to the other VMs in the VM group. The VM's previous CPU quota is relinquished and replaced by this member CPU quota (see [page 176](#)).

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
2	0	CMD0001	Decimal specifications rounded to 2 decimal places
	64	VMS4000	VM not yet created
	64	VMS4400	VM group not yet created
	64	VMS4422	VM not assigned to the CPU pool of the VM group
	64	VMS4424	VM already belongs to a VM group

Guaranteed messages

VMS4425

Example

```
/add-vm-to-vm-group vm-id=c1sys1,vm-group-name=gcust1,  
                    member-cpu-quota=1 _____ (1)  
VMS4425 VIRTUAL MACHINE (2,C1SYS1) ASSIGNED TO VM GROUP 'GCUST1'
```

- (1) The customer VM `C1SYS1` is added to the VM group `GCUST1`. The member CPU quota selected is 1; the CPU quota which the VM was given when it was created is relinquished.

See also the detailed [“Example of operating VM groups” on page 182](#).

ASSIGN-VM-GROUP-TO-CPU-POOL

Assign a VM group to a CPU pool (S servers)

SQ No VM groups are available on SQ servers. The command is rejected with message VMS3011 and with general return code VMS9004.

/ASSIGN-VM-GROUP-TO-CPU-POOL is used to assign a VM group to a (different) CPU pool. Together with the VM group, the VMs which are assigned to the VM group are also assigned to the (different) CPU pool.

This command is only permitted for the VM2000 administrator.

The VM group is assigned a (new) CPU quota.

ASSIGN-VM-GROUP-TO-CPU-POOL

```

VM-GROUP-NAME = <name 1..8>
,TO-CPU-POOL = *STDPOOL / <name 1..8>
,CPU-QUOTA = <fixed 0.01..99.99>
,ATTACHED-VM-CPUS = *CHECK-NUMBER / *ADJUST-NUMBER

```

VM-GROUP-NAME = <name 1..8>

Name of the VM group.

TO-CPU-POOL = *STDPOOL / <name 1..8>

Name of the VM group.

CPU-QUOTA = <fixed 0.01..99.99>

CPU quota of the VM group. Determines the VM group's own CPU share in comparison to the other (individual) VMs and VM groups which are assigned to the CPU pool (TO-CPU-POOL operand) (see [page 238](#)).

ATTACHED-VM-CPUS =

Determines command execution and the implicit actions of VM2000 for the virtual CPUs of all VMs of the VM group.

ATTACHED-VM-CPUS = *CHECK-NUMBER

The command is executed when the number of attached virtual CPUs of every VM in the VM group is less than or equal to the number of attached real CPUs in the CPU pool to which the VM group is to be assigned, see [page 228](#). If a VM in the VM group does not satisfy this condition, the command is rejected.

ATTACHED-VM-CPUS = *ADJUST-NUMBER

The command is executed unless a VM in the VM group is in the RUNNING or IN HOLD (WAIT) status and the CPU pool to which the VM group is to be assigned contains no attached real CPU.

With this operand value, VM2000 initiates asynchronous actions in the following cases:

- The number of attached virtual CPUs in a VM in the VM group is greater than the number of attached real CPUs in the CPU pool to which the VM group is assigned:
VM2000 detaches the excess virtual CPUs (implicit /DETACH-DEVICE in the BS2000 guest system for a corresponding number of virtual CPUs). These detached CPUs cannot initially be attached again by the guest system; they are identified accordingly by VM2000 (status of the virtual CPU in the VM2000 information commands: BLOCK).
- The number of attached virtual CPUs in a VM in the VM group is less than the number of attached real CPUs in the CPU pool to which the VM group is assigned **and** the number of attached virtual CPUs in this VM in the VM group is less than its multiprocessor level:

VM2000 cancels the BLOCK status for the required number of virtual CPUs of the VM and attaches them (implicit /ATTACH-DEVICE in the BS2000 guest system for the virtual CPU).

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
2	0	CMD0001	Decimal specifications rounded to 2 decimal places
5	0	CMD0001	Implicit reconfiguration started for virtual CPU(s)
	64	VMS4400	VM group not yet created
	64	VMS4600	CPU pool not yet created
	64	VMS4612	Too few attached CPUs added to the CPU pool
	64	VMS4626	VM group is already assigned to the CPU pool

Guaranteed messages

VMS4620, VMS4621, VMS4627

Example

```
/assign-vm-group-to-cpu-pool vm-group-name=extgroup,to-cpu-pool=exclpool,
    cpu-quota=1,attached-vm-cpus=*check-number
VMS4627 CPU POOL ASSIGNMENT OF VM GROUP 'EXTGROUP' CHANGED FROM
    CPU POOL '*STDPOOL' TO CPU POOL 'EXCLPOOL'
```

ASSIGN-VM-TO-CPU-POOL

Assign a VM to a CPU pool

`/ASSIGN-VM-TO-CPU-POOL` is used to assign a VM which does not belong to a VM group to a (different) CPU pool. The VM is assigned a (new) CPU quota.

This command is only permitted for the VM2000 administrator.

- SQ** On SQ servers the command is not executed if the CPU pool to which the VM is to be assigned contains no attached CPU.
 On SQ servers a VM cannot be assigned to another CPU pool while it is being migrated (Live Migration).

ASSIGN-VM-TO-CPU-POOL

```

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>
,TO-CPU-POOL = *STDPOOL / <name 1..8>
,CPU-QUOTA = <fixed 0.01..99.99>
,ATTACHED-VM-CPUS = *CHECK-NUMBER / *ADJUST-NUMBER
  
```

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>

Identifies the virtual machine by means of a VM index or VM name.

TO-CPU-POOL = *STDPOOL / <name 1..8>

Name of the CPU pool to which the VM is to be assigned.

CPU-QUOTA = <fixed 0.01..99.99>

CPU quota of the VM. Determines the VM's own CPU share in comparison to the other VMs or VM groups which are assigned to the CPU pool (`TO-CPU-POOL` operand) (see [page 238](#)).

ATTACHED-VM-CPUS =

Determines command execution and the implicit actions of VM2000 for the virtual CPUs of the VM.

ATTACHED-VM-CPUS = *CHECK-NUMBER

The command is executed if the number of attached virtual CPUs in the VM is less than the number of attached real CPUs in the CPU pool to which the VM is to be assigned, see [page 228](#). Otherwise the command will be rejected.

ATTACHED-VM-CPUS = *ADJUST-NUMBER

The command is executed unless the VM is in the RUNNING or IN HOLD (WAIT) status and the CPU pool to which the VM is to be assigned contains no attached real CPU.

With this operand value, VM2000 initiates asynchronous actions in the following cases:

- The number of attached virtual CPUs in the VM is greater than the number of attached real CPUs in the CPU pool to which the VM is assigned:

VM2000 detaches the excess virtual CPUs (implicit /DETACH-DEVICE in the BS2000 guest system for a corresponding number of virtual CPUs). These detached CPUs cannot initially be attached again by the guest system; they are identified accordingly by VM2000 (status of the virtual CPU in the VM2000 information commands: BLOCK).

- The number of attached virtual CPUs in the VM is less than the number of attached real CPUs in the CPU pool to which the VM is assigned, **and** the number of attached virtual CPUs in the VM is less than its multiprocessor level:

VM2000 cancels the BLOCK status for the required number of virtual CPUs of the VM and attaches them (implicit /ATTACH-DEVICE in the BS2000 guest system for the virtual CPU).

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
2	0	CMD0001	Decimal specifications rounded to 2 decimal places
3	0	CMD0001	Implicit attach started for virtual CPU(s)
4	0	CMD0001	Implicit detach started for virtual CPU(s)
	64	VMS4000	VM not yet created
	64	VMS4600	CPU pool not yet created
	64	VMS4612	Too few attached CPUs added to the CPU pool
	64	VMS4616	VM already assigned to the CPU pool
	64	VMS4622	VM belongs to a VM group
	128	VMS3042	VM being migrated. Command rejected

Guaranteed messages

VMS4618, VMS4620, VMS4621

Example

```
/assign-vm-to-cpu-pool vm-id=c1sys1,to-cpu-pool=pcust1,cpu-quota=80
VMS4618 CPU POOL ASSIGNMENT OF VIRTUAL MACHINE (2,C1SYS1) CHANGED
      FROM CPU POOL '*STDPOOL' TO CPU POOL 'PCUST1'
```

See also the detailed [“Example of managing CPU pools” on page 233](#).

ATTACH-VM-RESOURCES

Attach real CPUs

`/ATTACH-VM-RESOURCES` is used to attach detached real normal CPUs and extra CPUs for VM2000 operation.

This command is only permitted for the VM2000 administrator.

ATTACH-VM-RESOURCES
<pre> UNIT = *CPU(...) / *EXTRA-CPU(...) *CPU(...) CPU-IDENTIFICATION = *ANY(...) / list-poss(63): <x-text 1..2> *ANY(...) CPU-POOL-NAME = *UNIQUE / *STDPOOL / <name 1..8> *EXTRA-CPU(...) CPU-IDENTIFICATION = *ALL(...) / *ANY(...) / list-poss(63): <x-text 1..2> *ALL(...) CPU-POOL-NAME = *UNIQUE / *STDPOOL / <name 1..8> / *ALL *ANY(...) CPU-POOL-NAME = *UNIQUE / *STDPOOL / <name 1..8> </pre>

UNIT = *CPU(...)

Specifies real normal CPUs to be attached.



The attachment of a real processor has no effect on the reconfiguration state of the virtual CPUs of a VM. Virtual CPUs may be attached in the BS2000 guest system by means of the BS2000 command `/ATTACH-DEVICE UNIT=*CPU(CPU-IDENTIFIER=<mn>)`.

CPU-IDENTIFICATION = *ANY(...)

Selects any real normal CPU from a CPU pool.

CPU-POOL-NAME =

Determines the CPU pool.

CPU-POOL-NAME = *UNIQUE

A real normal CPU from the only existing CPU pool (*STDPOOL) is selected (operand provided for reasons of compatibility with earlier VM2000 versions).

CPU-POOL-NAME = *STDPOOL / <name 1..8>

Name of the CPU pool.

CPU-IDENTIFICATION = list-poss(63): <x-text 1..2>

Identifier of the real normal CPU.

UNIT = *EXTRA-CPU(...)

Specifies extra CPUs to be attached, see [page 222](#).

CPU-IDENTIFICATION = *ALL(...)

Selects all extra CPUs from a CPU pool.

CPU-POOL-NAME =

Determines the CPU pool.

CPU-POOL-NAME = *UNIQUE

All extra CPUs from the only existing CPU pool (*STDPOOL) are selected (operand provided for reasons of compatibility with earlier VM2000 versions).

CPU-POOL-NAME = *STDPOOL / <name 1..8>

Name of the CPU pool.

CPU-POOL-NAME = *ALL

All extra CPUs from all CPU pools are selected.

CPU-IDENTIFICATION = *ANY(...)

Selects any extra CPU from a CPU pool.

CPU-POOL-NAME =

Determines the CPU pool.

CPU-POOL-NAME = *UNIQUE

An extra CPU from the only existing CPU pool (*STDPOOL) is selected (operand provided for reasons of compatibility with earlier VM2000 versions).

CPU-POOL-NAME = *STDPOOL / <name 1..8>

Name of the CPU pool.

CPU-IDENTIFICATION = list-poss(63): <x-text 1..2>

Identifier of the extra CPU.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
1	0	CMD0001	No action required
	64	VMS3002	Invalid operand value (operand CPU-ID)
	64	VMS4043	CPU not known in system
	64	VMS4072	CPU defective
	64	VMS4073	CPU not attached because of CPU attribute
	64	VMS4074	No additional (extra) CPU available
	64	VMS4150	CPU could not be attached because of its state
	64	VMS4152	CPU is OFFLINE
	64	VMS4270	No extra CPU in the configuration
	64	VMS4600	CPU pool not yet created
	64	VMS4609	Several CPU pools created
	64	VMS4611	Selected CPUs do not belong to CPU pool
	64	VMS9005	Command not or only partially processed

Guaranteed messages

VMS4043, VMS4044, VMS4046, VMS4072, VMS4073, VMS4075, VMS4150, VMS4152

Example

```

/show-vm-resources information=*cpu _____ (1)
VM-ID      CPU-POOL  GR-CPU-Q  CPU-Q/  EFF-Q  MAX-U/  EFF-U  PROC
 1 MONITOR  *STDPOOL          14.00/  50.00  100.00/  50.00    2

VM-ID      SCHED  ACT-I  VIRT  CPUS
 1 MONITOR  TS     NO    00(RUN),01(RUN)

CPU-ATTR   CPU-POOL      REAL CPUS
NORMAL     *STDPOOL      0A(ATT),0B(ATT),0C(ATT),0D(DET)
EXTRA      *STDPOOL      0E(DET),0F(DET) _____ (2)
SPARE      NONE
FOREIGN    *POOL0        00(ATT),01(ATT),02(ATT),03(ATT)
           *FOREIGN    04(ATT),05(ATT),06(ATT),07(ATT),
           08(ATT),09(ATT)

/attach-vm-resources unit=*cpu(cpu-id=0d) _____ (3)
VMS4044 CPU 0D ATTACHED

/attach-vm-resources unit=*extra-cpu(cpu-id=*all) _____ (4)
VMS4075 EXTRA-CPU 0E ATTACHED
VMS4075 EXTRA-CPU 0F ATTACHED

```

```

/show-vm-resources information=*cpu _____ (5)
VM-ID      CPU-POOL  GR-CPU-Q  CPU-Q/  EFF-Q  MAX-U/  EFF-U  PROC
1 MONITOR  *STDPool          14.00/  50.00  100.00/ 50.00   2

VM-ID      SCHED  ACT-I  VIRT  CPUS
1 MONITOR  TS     NO    00(RUN),01(RUN)

CPU-ATTR   CPU-POOL      REAL CPUS
NORMAL     *STDPool      0A(ATT),0B(ATT),0C(ATT),0D(ATT)
EXTRA      *STDPool      0E(ATT),0F(ATT) _____ (6)
SPARE
FOREIGN    *POOL0        00(ATT),01(ATT),02(ATT),03(ATT)
           *FOREIGN     04(ATT),05(ATT),06(ATT),07(ATT),
           08(ATT),09(ATT)

```

- (1) The VM2000 administrator requests information on the CPU state.
- (2) The real normal CPU 0D and the extra CPUs are detached.
- (3) The normal CPU 0D is attached.
- (4) All the extra CPUs are attached.
- (5) The VM2000 administrator requests information on the CPU state.
- (6) All real normal CPU and all extra CPUs are now attached.

See also the detailed [“Example of managing CPU pools” on page 233](#).

BEGIN-VM-DIALOG

Start administration or operation

`/BEGIN-VM-DIALOG` is used to open a dialog from a privileged user task or after a connection has been established to `$VMCONS`,

- for administering the VM using VM2000 commands (ADMIN dialog).
Up to 15 ADMIN dialogs can be conducted concurrently for each VM from different, privileged user tasks. In addition, one ADMIN dialog can be conducted via `$VMCONS`.
- for operating the guest system via `$VMCONS` (VC dialog).
Up to eight VC dialogs can be conducted concurrently on each VM via `$VMCONS`.

Guest system operation with BS2000 consoles is described on [page 100](#).

`/BEGIN-VM-DIALOG` cannot be entered at a monitor system BS2000 console. Open dialogs are terminated by means of `/END-VM-DIALOG`, `/DELETE-VM`, `/LOGOFF` of the user task or when the connection to `$VMCONS` is terminated.

BEGIN-VM-DIALOG

```

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>
, PASSWORD = *NONE / <c-string 1..8> / <x-string 1..16>
, TYPE = *STD / *BOTH / *ADMIN(...) / *VC(...)
    *ADMIN(...) / *VC(...)
    |   OUTPUT-PREFIX = *YES / *NO

```

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>

Identifies the virtual machine by means of a VM index or VM name.

PASSWORD = *NONE / <c-string 1..8> / <x-string 1..16>

Password defined by means of `/CREATE-VM` or `/MODIFY-VM-ATTRIBUTES`.

TYPE =

Defines the type of dialog to be opened for command input via a \$VMCONS connection. This operand is ignored with command input from a privileged task, as in this case only one ADMIN dialog (without output prefix) can be started.

TYPE = *STD

The following applies in a user task: an ADMIN dialog is to be started.

The following applies for a \$VMCONS connection:

[S] An ADMIN dialog and a VC dialog are to be started.

[SQ] An ADMIN dialog is to be started.

TYPE = *BOTH

Both an ADMIN dialog and a VC dialog are to be opened. The VM prefix or GS prefix are attached to the outputs from the ADMIN or VC dialog.

TYPE = *ADMIN(...)

Operation of the VM using VM2000 commands (ADMIN dialog).

OUTPUT-PREFIX = *YES / *NO

Specifies whether the VM prefix VMnn is attached to the outputs of the ADMIN dialog.

TYPE = *VC(...)

Operation of the guest system (VC dialog).

OUTPUT-PREFIX = *YES / *NO

Specifies whether the GS prefix GSnn is attached to the outputs of the VC dialog.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
1	0	CMD0001	No action required
	64	VMS1000	You have already opened an ADMIN dialog
	64	VMS1001	ADMIN dialog already opened for the VM with another partner
	64	VMS1024	Invalid password specified
	64	VMS1035	Maximum number of VC dialogs for the VM has been reached
	64	VMS1050	A VC dialog with KVP could not be configured
	64	VMS1522	Maximum number of ADMIN/VC dialogs has been reached
	64	VMS2101	VM2000 shutdown in progress
	64	VMS4000	VM not yet created

Example (privileged user task)

```
/begin-vm-dialog vm-id=system2,password=c'sys2',type=*std _____ (1)
```

(1) An ADMIN dialog is opened from a privileged user task.

Example (via \$VMCONS)

```
vm02:/begin-vm-dialog vm-id=system2,password=c'sys2',type=*both _____ (1)
```

```
VM01:VMS1013 PARTNER 'OMS00012 SHIRKVM1' HAS STARTED AN ADMINISTRATION DIALOG  
WITH VIRTUAL MACHINE (2,SYSTEM2). ADMINISTRATION-ID USED '@002' _____ (2)
```

```
VM01:VMS1033 PARTNER 'OMS00012 SHIRKVM1' HAS STARTED A VC DIALOG WITH  
VIRTUAL MACHINE (2,SYSTEM2)
```

```
VM02:VMS3000 VM2000 COMMAND /BEGIN-VM-DIALOG FOR  
VIRTUAL MACHINE (2,SYSTEM2) PROCESSED SUCCESSFULLY
```

```
VM02:NBRO740 COMMAND COMPLETED 'BEGIN-VM-DIALOG';  
RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
```

- (1) After a connection has been established to \$VMCONS, the ADMIN dialog is opened along with a VC dialog with VM2.
- (2) The VM2000 administrator is informed of the start of the ADMIN dialog and the VC dialog.

CALL-VM-PROCEDURE

Start a VM2000 procedure file

`/CALL-VM-PROCEDURE` is used to start a VM2000 procedure file in which a sequence of VM2000 commands is stored.

All VM2000 commands can be used in the procedure file except the following: `/CALL-VM-PROCEDURE`, `/BEGIN-VM-DIALOG` and `/END-VM-DIALOG`.

The following BS2000 commands are permissible:

- `/STEP`
prevents abortion of the procedure in the event of an error. The procedure is continued normally starting from the `/STEP` command.
- `/REMARK`
allows comments logged during the command sequence to be inserted in the procedure file.

If a command is more than one line long, the hyphen (-) is used as a continuation character. The hyphen can be positioned after any comma in the operand field and indicates that there is a continuation line. A hyphen may be followed only by blanks.

Continuation lines

If a command is more than one line long, the hyphen (-) is used as a continuation character. The hyphen can be positioned after any comma in the operand field and indicates that there is a continuation line. A hyphen may be followed only by blanks.

Each continuation line begins with a slash, followed by the rest of the operands of the command.

Commands in procedure files may have a maximum length of 300 characters, not including hyphens and fill characters (blanks).

A procedure containing commands with continuation lines can no longer be processed by means of `/RUN`.

Processing the VM2000 procedure file

`/CALL-VM-PROCEDURE` processes the commands sequentially. If one of the commands produces an error, processing will be continued with the next `/STEP`.

`/CALL-VM-PROCEDURE` can be entered only in an ADMIN dialog via `$VMCONS`. It can be entered not via a user task (where S or non-S procedures are available) or at a monitor system BS2000 console (where such a procedure file can be started using `/RUN`).

If the procedure file contains commands which are reserved for the VM2000 administrator, /CALL-VM-PROCEDURE must be entered in the ADMIN dialog with the **monitor VM**.

CALL-VM-PROCEDURE
<p>FILE-NAME = <filename 1..54 without-gen-vers> ,VM-IDENTIFICATION = *CURRENT / <integer 1..99> / <name 1..8> ,LIST = *YES / *NO</p>

FILE-NAME = <filename 1..54 without-gen-vers>

File name of the procedure file.

VM-IDENTIFICATION =

Identification of the virtual machine. The VM whose identification is specified must not be in the NOT INITIALIZED state.

VM-IDENTIFICATION = *CURRENT

The VM identification assigned previously by means of /BEGIN-VM-DIALOG is used.

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>

Identifies the virtual machine by means of a VM index or VM name.



Here the significance of the VM-IDENTIFICATION operand is quite different to its meaning in other VM2000 commands. In all commands this operand value replaces the default (VM-IDENTIFICATION=*CURRENT or VM-IDENTIFICATION not specified) in the procedure file for which the VM-IDENTIFICATION operand is **optional**.

In the case of all commands in the procedure file where the VM-IDENTIFICATION operand is **mandatory**, the VM-ID is not replaced.

If /CREATE-VM is used in a procedure file where no specification has been made for the VM-INDEX and VM-NAME operands, they will not be replaced.

LIST =

Defines the type of logging for procedure execution.

LIST = *YES

When the procedure is executed, procedure file commands and messages are logged to the VM terminal.

LIST = *NO

Logging of the commands and messages is suppressed. Only those messages relating to errors which occurred during processing of the procedure are output.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
	64	VMS1506	Procedure file is not of type SAM or ISAM
	64	VMS1562	Error on opening procedure file
	64	VMS3010	Illegal command in procedure file
	64	VMS4000	VM not yet created
	128	VMS1505	Procedure mode already active for VM

Example of a procedure file

```

/REMARK TESTVM MEMORY=512 MBYTE _____ (1)
/CREATE-VM MEM=512,VM-NAME=TESTVM _____ (2)
/REMARK ADD DEVICES
/ADD-VM-DEVICES UNITS=(Z2,Z3,D0,D1),VM-NAME=TESTVM _____ (3)
/REMARK DIALOG-STARTUP ON TESTVM
/START-VM IPL-UNIT=D0,-
/INFORMATION-BYTE=*DIALOG,VM-IDENTIFICATION=TESTVM _____ (4)
/STEP _____ (5)
/REMARK TESTVM2 MEMORY=1024 MBYTE
/CREATE-VM MEM=1024,VM-NAME=TESTVM2 _____ (6)

```

- (1) Comment line.
- (2) A VM with the VM name TESTVM is initialized.
- (3) Devices are assigned to the VM with the VM name TESTVM.
- (4) The guest system on the VM is started. The command is input with a continuation line.
- (5) If any of the commands under (2), (3) or (4) cannot be executed, the procedure is continued from (5).
- (6) An additional VM with the VM name TESTVM2 is initialized.

CREATE-VM

Initialize a VM

`/CREATE-VM` initializes a VM, i.e. a VM is created under VM2000.

This command is only permitted for the VM2000 administrator.

During the initialization process, the following attributes and resources are assigned to the VM:

- VM index and VM name (identification of the VM)
- main memory size of the VM
- minimum and maximum main memory size of the VM for main memory reconfiguration
- location of the VM in the main memory of VM2000
- (member) CPU quota and maximum CPU utilization of the VM
- Maximum I/O utilization of the VM
- adding of the VM to a VM group
- assignment of the VM to a CPU pool
- multiprocessor level of the VM
- the password for administration
- Range of commands available to VM2000 and VM administrators
- Privileges of the VM



We recommend initializing a VM with `IO-PRIORITY=*NO` and `IO-RESET=*NO` (i.e. without these privileges) and only assigning the privileges with `/MODIFY-VM-ATTRIBUTES`.

- Settings for control over the real CPU
- Monitor job variable for monitoring the VM.

After the command has been executed successfully, the VM has the status `INIT-ONLY`. Any other `/CREATE-VM` commands for this VM are rejected.



On SQ servers a VM's firmware component is started when the VM is initialized. Despite the `INIT-ONLY` status, the VM already utilizes a minimal CPU capacity.



The monitor VM is automatically initialized when VM2000 is initialized. Its attributes and resources are configured when VM2000 is installed (see [chapter "Installing VM2000" on page 43](#)).

(part 1 of 2)

CREATE-VM

```

MEMORY-SIZE = <integer 1..1048576>(DIMENSION = *MEGABYTE / *GIGABYTE)
,MIN-MEMORY-SIZE = *STD / <integer 1..1048576>(DIMENSION = *MEGABYTE / *GIGABYTE)
,VM-NAME = *STD / <name 1..8>
,CPU-QUOTA = *STD / <fixed 0.01..99.99> / *BY-VM-GROUP(...)
    *BY-VM-GROUP(...)
        | VM-GROUP-NAME = <name 1..8>
        | ,MEMBER-CPU-QUOTA = *STD / <fixed 0.01..99.99>
,MAX-CPU-UTILIZATION = 100 / <fixed 0.01..100.00>
,MAX-IO-UTILIZATION = 100 / <integer 1..100>
,CPU-POOL-NAME = *STD / *STDPOOL / <name 1..8>
,PASSWORD = *NONE / <c-string 1..8> / <x-string 1..16>
,PROCESSOR = *MONO / *BI / *TRIPLE / *QUADRO / *OCTO / *MAXIMUM / *EXTRA-AND-NORMAL /
    <integer 1..32>
,LOCATION = *ANY / *BEFORE-VM(...) / *AFTER-VM(...) / *MEMORY-END / <x-text 1..8>(…)
    *BEFORE-VM(…)
        | VM-IDENTIFICATION = <integer 2..99> / <name 1..8>
    *AFTER-VM(…)
        | VM-IDENTIFICATION = <integer 1..99> / <name 1..8>
    <x-text 1..8>(…)
        | DIMENSION = *BYTE / *MEGABYTE
,VM-INDEX = *ANY / <integer 1..99>
,ADMIN-PRIVILEGES = *STD / *MINIMUM / *MAXIMUM
,PRIVILEGES = *PARAMETERS(…)
    *PARAMETERS(…)
        | IO-RESET = *NO / *YES
        | ,IO-PRIORITY = *NO / *YES
        | ,AUTO-SNAP-ASSIGNMENT = *YES / *NO
        | ,ASSIGN-BY-GUEST = *NO / *YES(…)
        | *YES(…)
            | SET-NAME = *STDSET / *ALL / *EXCEPT(…) / list-poss(8): <name 1..8> / *STDSET
            | *EXCEPT(…)
                | SET-NAME = list-poss(8): <name 1..8> / *STDSET


```


(part 2 of 2)

```
,VM-ACTIVE-IDLE= *NO / *AT-DEDICATED-CPUS
,MONJV= *NONE / <filename 1..54 without-gen-vers>
,JV-PASSWORD = *NONE / <c-string 1..4> / <x-string 1..8> / <integer -2147483647..2147483648>
,MAX-MEMORY-SIZE = *STD / <integer 1..1048576>(DIMENSION = *MEGABYTE / *GIGABYTE)
```

MEMORY-SIZE = <integer 1..1048576>(DIMENSION = *MEGABYTE / *GIGABYTE)


Size of main memory for the appropriate VM up to a size of 1 Tbyte (terabyte; 1 Tbyte = 1024 Gbytes = 1 048 576 Mbytes).

 On SQ servers a small amount of a VM's main memory is required for the firmware component. The main memory of a VM should be at least 512 Mbytes large, see [page 140](#). The value must be a multiple of 2 Mbytes.

 When MEMORY-SIZE > 32 Gbytes is specified for a guest system BS2000/OSD V6.0, the main memory of this guest system can only be reduced later to a value less than 32 Gbytes.

MIN-MEMORY-SIZE = *STD / <integer 1..1048576>(DIMENSION = *MEGABYTE / *GIGABYTE)

Minimum size of the main memory for the relevant VM when main memory is reconfigured. The default is the size of the main memory for the VM as defined by MEMORY-SIZE.

 On SQ servers the main memory of a VM should be at least 300 Mbytes in size, see [page 140](#). The value must be a multiple of 2 Mbytes.

VM-NAME =


Name of the new VM. This identifies the VM in addition to the VM index.

VM-NAME = *STD

Standard VM name. VM00nn is assigned as the standard name, nn being the VM index (nn=01..99).

VM-NAME = <name 1..8>

VM name assigned by the VM2000 administrator. A VM name corresponding to the standard name of another VM is not permissible (e.g VM-NAME=VM0002, VM-INDEX=5).

 See also the general conditions for SQ servers in section [“Recommendations for the definition and use of VM names”](#) on [page 139](#).

CPU-QUOTA = *STD / <fixed 0.01..99.99> / *BY-VM-GROUP(...)

CPU quota of the VM. The default value is 1 (the VM is not added to any VM group).

CPU-QUOTA = *STD / <fixed 0.01..99.99>

CPU quota of the VM which is not (yet) added to a VM group when it is created Determines the VM's own share of the CPU capacity of the CPU pool in comparison to the VM groups and the other VMs which have not been added to a VM group (see [page 238](#)).

CPU-QUOTA = *BY-VM-GROUP(...)

The VM is added to a VM at creation time (S servers).

VM-GROUP-NAME = <name 1..8>

Name of the VM group to which the VM is added.

MEMBER-CPU-QUOTA = *STD / <fixed 0.01..99.99>

Member CPU quota of the VM which is added to a VM group at creation time. This determines the VM's own CPU share in comparison to the other VMs in the VM group (see [page 176](#)) The default value is 1.

MAX-CPU-UTILIZATION = 100 / <fixed 0.01..100.00>

Maximum CPU utilization of the VM (see [page 238](#)).

- [S] If the VM belongs to a VM group, its maximum CPU utilization is also restricted by the "maximum CPU utilization of the VM group" (see [page 176](#)).

MAX-IO-UTILIZATION = 100 / <integer 1..100>

Maximum I/O utilization of the VM (S servers, see [page 360](#)).

CPU-POOL-NAME =

Determines the CPU pool to which the VM is assigned. Every VM is always assigned to precisely one CPU pool.

- [SQ] On SQ servers the CPU pool must possess at least one attached real CPU.



The CPU-QUOTA operand determines the (individual) VM's or VM group's own CPU share in comparison to the other VMs or VM groups which are assigned to the CPU pool (see [page 249](#)).

CPU-POOL-NAME = *STD

When an (individual) VM is initialized (CPU-QUOTA = *STD / <fixed 0.01..99.99>), it is assigned to the standard CPU pool.

- [S] When the VM is assigned to a VM group (CPU-QUOTA = *BY-VM-GROUP(...)), it is assigned to the CPU pool of the VM group.

CPU-POOL-NAME = *STDPOOL / <name 1..8>

Name of the CPU pool to which the VM is assigned.

- [S] If the VM is added to a VM group (CPU-QUOTA=*BY-VM-GROUP(. . .) operand), it may only be assigned to the CPU pool of the VM group.

PASSWORD =

Password controlling access authorization in /BEGIN-VM-DIALOG. The password must not exceed 8 bytes in length.

PASSWORD = *NONE

No password is necessary.

PASSWORD = <c-string 1..8>

Password in character form.

PASSWORD = <x-string 1..16>

Password in hexadecimal form. Passwords comprising zeros only (X'00..00') are ignored.

PROCESSOR =

This defines the multiprocessor level of the VM (see [page 215](#)).

PROCESSOR = *MONO

A VM with one virtual CPU is set up.

PROCESSOR = *BI

A VM with two virtual CPUs is set up.

PROCESSOR = *TRIPLE

A VM with three virtual CPUs is set up.

PROCESSOR = *QUADRO

A VM with four virtual CPUs is set up.

PROCESSOR = *OCTO

A VM with eight virtual CPUs is set up.

PROCESSOR = *MAXIMUM

A VM with the maximum possible number of virtual CPUs is created. The maximum possible number of virtual CPUs matches the number of real normal CPUs which are available for VM2000 operation, see [page 213](#).

- [S] On S servers the maximum number of virtual CPUs is 8.

PROCESSOR = *EXTRA-AND-NORMAL

A VM is configured with as many virtual CPUs as normal CPUs and extra CPUs are available for VM2000 operation.

[S] On S servers up to 8 virtual CPUs are possible.

A VM with this number of CPUs can utilize the maximum CPU capacity, including the CPU capacity of the attached extra CPUs. For a server without extra CPUs this value is the same as the value of `PROCESSOR=*MAXIMUM`.

PROCESSOR = <integer 1..32>

A VM is set up with the number of virtual CPUs specified here. A value less than or equal to the value of `PROCESSOR=*MAXIMUM` is permissible.

LOCATION =

Specifies the location of the VM in the main memory of VM2000 (S servers).

LOCATION = *ANY

The location is selected by VM2000.

LOCATION = *BEFORE-VM(VM-IDENTIFICATION = <integer 2..99> / <name 1..8>)

Specifies the VM before which the new VM is to be located.

LOCATION = *AFTER-VM(VM-IDENTIFICATION = <integer 1..99> / <name 1..8>)

Specifies the VM after which the new VM is to be located.

LOCATION = *MEMORY-END

The new VM is to be located at the end of the VM2000 main memory

LOCATION = <x-text 1..8>(DIMENSION = *BYTE / *MEGABYTE)

Specifies the address with the dimension. The address must be a multiple of 1 Mbyte. Permissible range of values: 00000000..7FF00000 (hexadecimal numbers). If required the address is padded to the left with zeros.

VM-INDEX =

The VM index.

VM-INDEX = *ANY

The VM index is assigned by VM2000.

VM-INDEX = <integer 1..99>

Specification of the VM index.

Highest VM index=15 (depending on the server model)

ADMIN-PRIVILEGES =

Defines the command/function range for the VM2000 administrator and the VM administrator (see [page 192](#)).

ADMIN-PRIVILEGES = *STD

Standard command/function range.

ADMIN-PRIVILEGES = *MINIMUM

Restricted command/function range for the VM2000 administrator.

- [S] The operand value *MINIMUM only be specified on S servers (/CREATE-VM in the configuration file of the monitor VM or when the monitor system is started up in dialog mode).
- [SQ] On SQ servers the command/functional scope for the VM2000 administrator can only be restricted with /MODIFY-VM-ATTRIBUTES.

ADMIN-PRIVILEGES = *MAXIMUM

Extended command/function range for the VM administrator. The operand value *MAXIMUM is rejected if specified for the monitor VM.

PRIVILEGES = *PARAMETERS(...)

Describes privileges of the VM.

IO-RESET = *NO / *YES

Specifies on S servers whether the channels are reset **in the hardware** for this VM with /START-VM or when the guest system is restarted or the channels reconfigured (see [page 145](#)).

IO-PRIORITY = *NO / *YES

Specifies on S servers whether a waiting virtual CPU of the VM is executed again on a real CPU by the VM2000 hypervisor immediately after the end of pending input/output for this VM (see [page 146](#)).



The total number of virtual CPUs of all the VMs with this privilege must not be greater than that of the real normal CPUs of the server (excluding spare and extra CPUs).

AUTO-SNAP-ASSIGNMENT = *YES / *NO

Determines whether the guest system on the VM may implicitly assign itself snap units for Snapsets without the VM and device being assigned the ASSIGN-BY-GUEST privilege or attribute.

ASSIGN-BY-GUEST = *NO / *YES(...)

Specifies whether devices can also be implicitly assigned to this VM by the guest system (*YES). Alternatively, (*NO) the devices must only be assigned to the VM by the VM or VM2000 administrator. See [page 146](#).

SET-NAME =

Determines the assignment sets from which the guest system may implicitly assign devices to the VM.

SET-NAME = *STDSET / list-poss (8): <name 1..8> / *STDSET

Explicit specification of the assignment sets.

SET-NAME = *ALL

All existing and future assignment sets.

SET-NAME = *EXCEPT(SET-NAME = list-poss (8): <name 1..8> / *STDSET)

All existing and future assignment sets with the exception of those which are specified with SET-NAME.

VM-ACTIVE-IDLE =

Determines on S servers whether, in the event of fixed CPU assignment (dedicated CPUs), a VM still retains control over a real CPU if the VM's virtual CPU which runs on this is inactive (interruptible wait state), see [page 147](#).

VM-ACTIVE-IDLE = *NO

The VM2000 hypervisor deletes the real CPU assigned if the VM's virtual CPU which runs on this is inactive (interruptible wait state, "Idle").

VM-ACTIVE-IDLE = *AT-DEDICATED-CPUS

The VM retains control over the real CPU even if the VM's virtual CPU which runs on this is inactive (interruptible wait state, "Idle"). This idle time is then indicated in the VM2000 accounting records, with /SHOW-VM-STATUS (VM-ACTIVE output column) and in the VM2000 report of openSM2 as a time in which the VM actively uses the real CPU.



With fixed CPU assignment, VM-ACTIVE-IDLE=*AT-DEDICATED-CPUS is only effective if the VM's maximum CPU utilization (see [page 142](#)) is not restricted.

MONJV =

Defines a MONJV for monitoring the VM (see [page 148](#)).

MONJV = *NONE

No MONJV is defined.

MONJV = <filename 1..54 without-gen-vers>

The specified MONJV is supplied with data by VM2000; the sender of the command must therefore have write permission for the specified MONJV. If no catalog ID or user ID is specified in the file name, VM2000 completes the file name, in accordance with the runtime environment of the sender of the command. If a file name which does not yet have a job variable is specified, VM2000 creates the MONJV as shareable and without a password.

JV-PASSWORD = *NONE / <c-string 1..4> / <x-string 1..8> / <integer ...>

Password of the MONJV for checking access authorization. The input field is automatically blanked in the guided dialog. The value input is not logged.



This operand is only evaluated if a previously created job variable was specified with MONJV.

MAX-MEMORY-SIZE = *STD / <integer 1..1048576>(DIMENSION = *MEGABYTE / *GIGABYTE)

Maximum size of the main memory for the VM when main memory is reconfigured. The default value is twice the size of the VM's main memory which is predefined by MEMORY-SIZE. Take note of the information on [page 141](#).



This operand has no effect for S servers.



The value must be a multiple of 2 Mbytes for SQ servers.

Command return code, see [The uniqueness of the VM name in the HA cluster cannot currently be checked, see page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
1	0	CMD0001	No action required
2	0	CMD0001	Decimal specifications rounded to 2 decimal places
	64	VMS2101	VM2000 shutdown in progress
	64	VMS3002	Invalid operand value
	64	VMS3034	ADMIN-PRIVILEGES=*MIN only permissible for the monitor VM
	64	VMS3035	IO-PRIORITY cannot be set (number of virtual CPUs with a privilege too high)
	64	VMS4000	VM not yet created (*BEFORE-VM or *AFTER-VM operand)
	64	VMS4010	Maximum permissible number of virtual machines running in parallel has already been reached
	64	VMS4011	Requested memory size or location not available
	64	VMS4012	Requested memory size too small
	64	VMS4013	VM name not available
	64	VMS4014	SVP error
	64	VMS4015	VM ID already reserved
	64	VMS4071	Invalid memory size or location
	64	VMS4093	Invalid memory size
	64	VMS4200	Assignment set not yet created
	64	VMS4340	MONJV could not be accessed
	64	VMS4341	MONJV already assigned to another VM
	64	VMS4400	VM group not yet created
	64	VMS4422	VM not assigned to the CPU pool of the VM group
	64	VMS4600	CPU pool not yet created
	64	VMS4612	No attached CPU is assigned to the CPU pool (SQ servers)
	64	VMS9004	Functionality not supported on current HSI or with current parameter assignment
	130	VMS2202	The uniqueness of the VM names in the HA cluster cannot currently be checked

Guaranteed messages

VMS4425, VMS4615

Example

```

/create-vm memory-size=1024,vm-name=system2,cpu-quota=30,
           processor=*mono,password=c'sys2',monjv=jv-sys2 _____ (1)
VMS3050 VIRTUAL MACHINE (2,SYSTEM2) CREATED _____ (2)
VMS4615 VIRTUAL MACHINE (2,SYSTEM2) ASSIGNED TO CPU POOL '*STDPOOL'
/show-vm-attributes vm-id=system2 _____ (3)
           CPU-POOL(P)
VM-ID      VM-GROUP(G) CPU-Q   MAX-U   PROC   MEM-SIZE  MAX-IO
 2 SYSTEM2 *STDPOOL(P) 30.00 100.00   1     1024 MB (100)

VM-ID      STATE              GUEST  VERSION ADMIN PRIV
 2 SYSTEM2 INIT-ONLY              UNKNOWN        NO   AS
/show-vm-attributes vm-id=system2,information=*monjv _____ (4)
VM-ID      VM-MONJV
 2 SYSTEM2 :4IVR:$TSOS.JV-SYS2

/show-jv jv-contents=jv-sys2 _____ (5)
$I I 0     4IVR     V0812011-11-26092636                SYSTEM2
002NONE

```

- (1) A VM is initialized by the VM2000 administrator. The VM receives the VM name SYSTEM2. It is created as a mono-VM. The size of main memory of the VM is 1024 Mbytes, and the CPU quota is 30. The password SYS2 is defined for dialog access. The MONJV jv-sys2 is used to monitor the VM.
- (2) VMVM2000 reports the initialization of the VM. The VM is assigned to the standard CPU pool.
- (3) The VM2000 administrator requests the attributes and status of VM2.
- (4) The VM2000 administrator queries the complete name of the MONJV.
- (5) The BS2000 command shows the contents of the MONJV (Column 1: Status \$I, Column 17: Identifier V); the guest system is not loaded (Columns 82-86: NONE).

See also the detailed [“Example of operating VM groups” on page 182](#) and the detailed [“Example of managing CPU pools” on page 233](#).

CREATE-VM-ASSIGNMENT-SET

Create assignment set

`/CREATE-VM-ASSIGNMENT-SET` is used to create an assignment set in VM2000. Up to 254 assignment sets can be created. The assignment set `*STDSET` is created by VM2000 when VM2000 operation is initialized.

This command is only permitted for the VM2000 administrator.

Devices can be assigned to a created assignment set using `/MODIFY-VM-DEVICE-ATTRIBUTES`. VMs can be assigned privileges for this assignment set using `/CREATE-VM` or `/MODIFY-VM-ATTRIBUTES`.

CREATE-VM-ASSIGNMENT-SET
SET-NAME = <name 1..8>

SET-NAME = <name 1..8>

Name of the new assignment set.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
	64	VMS4202	Assignment set already created
	64	VMS4210	Maximum number of assignment sets already created

Guaranteed messages

VMS4201

Example

```
/create-vm-assignment-set set-name=sys2impl
VMS4201 ASSIGNMENT SET 'SYS2IMPL' CREATED
```

See also the detailed examples for device and pubset management on [page 294](#) and on [page 324](#).

CREATE-VM-CPU-POOL

Create a CPU pool

`/CREATE-VM-CPU-POOL` is used to create a CPU pool under VM2000.

This command is only permitted for the VM2000 administrator.

CPUs, VMs and VM groups can be assigned to a created CPU pool.

CREATE-VM-CPU-POOL
CPU-POOL-NAME = <name 1..8>

CPU-POOL-NAME = <name 1..8>

Name of the new CPU pool.

SQ On SQ servers the characters #, \$ and @ should not be used in the CPU pool name.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
	64	VMS4602	CPU pool already created
	64	VMS4610	Maximum number of CPU pools already created

Guaranteed messages

VMS4601

Example

```
/create-vm-cpu-pool cpu-pool-name=pcust1
VMS4601 CPU POOL 'PCUST1' CREATED
```

See also the detailed [“Example of managing CPU pools” on page 233](#).

CREATE-VM-GROUP

Create VM group (S servers)

SQ No VM groups are available on SQ servers. The command is rejected with message VMS3011 and with general return code VMS9004.

/CREATE-VM-GROUP is used to create a VM group under VM2000.

This command is only permitted for the VM2000 administrator.

When a VM group is created it is assigned the following attributes:

- Name of the VM group
- CPU quota of the VM group
- Maximum CPU utilization of the VM group
- CPU pool, to which the VM group is assigned

VMs can be assigned to a created VM group.

CREATE-VM-GROUP

```

VM-GROUP-NAME = <name 1..8>
, CPU-QUOTA = *STD / <fixed 0.01..99.99>
, MAX-CPU-UTILIZATION = 100 / <fixed 0.01..100.00>
, CPU-POOL-NAME = *STDPOOL / <name 1..8>

```

VM-GROUP-NAME = <name 1..8>

The name of the new VM group.

CPU-QUOTA = *STD / <fixed 0.01..99.99>

CPU quota of the VM group. Determines the CPU share of the VM group in comparison to other VM groups and VMs which do not belong to any VM group (see [page 176](#)). The default value is 1.

MAX-CPU-UTILIZATION = 100 / <fixed 0.01..100.00>

Maximum CPU utilization of the VM group, see [page 176](#). This value also restricts the maximum CPU utilization of each VM which is added to the VM group.

CPU-POOL-NAME = *STDPOOL / <name 1..8>

Name of the CPU pool to which the VM group is assigned. Each VM group is always assigned to precisely one CPU pool.



The CPU-QUOTA operand determines the VM groups's own CPU share in comparison to the other (individual) VMs and VM groups which are assigned to the CPU pool (see [page 249](#)).

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
2	0	CMD0001	Decimal specifications rounded to 2 decimal places
	64	VMS4610	VM group already created
	64	VMS4410	Maximum number of VM groups already reached
	64	VMS4600	CPU pool not yet created

Guaranteed messages

VMS4401, VMS4625

Example

```
/create-vm-group vm-group-name=gcust1,cpu-quota=20,
                 max-cpu-utilization=20,cpu-pool-name=*stdpool _____ (1)
VMS4401 VM GROUP 'GCUST1' CREATED
VMS4625 VM GROUP 'GCUST1' ASSIGNED TO CPU POOL '*STDPOOL'
```

- (1) The VM2000 administrator creates the VM group GCUST1 and assigns it a CPU quota and a maximum CPU utilization of 20. VM group GCUST1 is assigned to the standard CPU pool.

See also the detailed [“Example of operating VM groups” on page 182](#).

DELETE-VM

Terminate a VM

One or more VMs are terminated using /DELETE-VM. The device assignment, the assignment to a CPU pool, the affiliation to a VM group and initialization of the VM are canceled.

The monitor VM cannot be terminated by this command.

[SQ] On SQ servers the main memory is released asynchronously and this can take several seconds. Only then is the released memory available for other or new VMs.

When the /DELETE-VM command is issued, all ADMIN dialogs with the VM and all VC dialogs with the guest system on the VM are terminated. The output of periodic monitored data (/SHOW-VM-STATUS PERIOD=<integer>) for the VM is also terminated.

After /DELETE-VM has been executed, the VM is in the NOT INITIALIZED state.

[i] /DELETE-VM . . . ,CHECK-VM-STATE=*NO terminates an active guest system immediately. As a result, files of the guest system which is running on the VM may be inconsistent. To prevent this from happening, a shutdown should be executed before the VM in the guest system is terminated.

If /DELETE-VM could not be executed, the VM remains in the state IN HOLD. It is not normally possible to resume guest system execution with /RESUME-VM, as important resources may already have been removed from the VM.

DELETE-VM

VM-IDENTIFICATION = ***CURRENT** / ***ALL** / <integer 1..99> / <name 1..8>
 ,CHECK-VM-STATE = ***YES** / ***NO**

VM-IDENTIFICATION =
 Identification of the virtual machine.

VM-IDENTIFICATION = *CURRENT
 The VM identification assigned previously by means of /BEGIN-VM-DIALOG is used.

VM-IDENTIFICATION = *ALL
 All VMs are to be terminated.
 This operand is only permitted for the VM2000 administrator.

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>
 Identifies the virtual machine by means of a VM index or VM name.

CHECK-VM-STATE =

Determines whether the command is executed, depending on the current state of the VM.

CHECK-VM-STATE = *YES

The command is only executed if the VM is in the DOWN or INIT-ONLY state.

[SQ] On SQ servers a VM cannot be terminated with CHECK-VM-STATE=*YES while it is being migrated (Live Migration).

CHECK-VM-STATE = *NO

The command is executed regardless of the VM state.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
1	0	CMD0001	Command successfully executed
	0	CMD0001	No action required
	64	VMS3040	Not processed because of the state of the VM
	64	VMS4000	VM not yet created
	64	VMS4002	VM could not be terminated
	64	VMS9005	Command not or only partially processed
	128	VMS3042	VM being migrated. Command rejected

Guaranteed messages

VMS3041, VMS4002, VMS4024, VMS4429, VMS4619 and the guaranteed messages for /REMOVE-VM-DEVICES.

Example

```

/delete-vm vm-id=system2,check-vm-state=*no ----- (1)
VMS4009 DEVICE 'CG' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4009 DEVICE 'CH' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4009 DEVICE '505A' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4009 DEVICE '505F' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4009 DEVICE '5060' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4619 VIRTUAL MACHINE (2,SYSTEM2) REMOVED FROM CPU POOL '*STDPOOL'
VMS4024 VIRTUAL MACHINE (2,SYSTEM2) TERMINATED ----- (2)

```

- (1) The VM with the name SYSTEM2 is terminated by the VM2000 administrator without regard for the system state.
- (2) VM2000 confirms termination of VM2.

DELETE-VM-ASSIGNMENT-SET

Deletes an assignment set

/DELETE-VM-ASSIGNMENT-SET is used to delete an assignment set.

This command is only permitted for the VM2000 administrator.

The following conditions must be satisfied:

- No further device may be assigned to the assignment set.
- The assignment set may not be in use for new devices (/MODIFY-VM-DEVICE-ATTRIBUTES UNITS=*NEW-DEVICES).
- The assignment set may no longer be assigned to a VM (ASSIGN-BY-GUEST=*YES(SET-NAME=...) privilege).

The assignment set *STDSET cannot be deleted.

The EXCEPT list of the assignment sets for a VM will be reduced if required.

DELETE-VM-ASSIGNMENT-SET
SET-NAME = <name 1..8>

SET-NAME = <name 1..8>

Name of the assignment set.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
	64	VMS4200	Assignment set not yet created
	64	VMS4204	Assignment set is still in use

Guaranteed messages

VMS4205

Example

```
/delete-assignment-set set-name=sys2impl
VMS4205 ASSIGNMENT SET 'SYS2IMPL' DELETED
```

See also the detailed examples for device and pubset management on [page 294](#) and on [page 324](#).

DELETE-VM-CPU-POOL

Delete CPU pool

/DELETE-VM-CPU-POOL is used to delete a CPU pool.

This command is only permitted for the VM2000 administrator.

No CPUs may still be contained in the CPU pool. No (single) VMs or VM groups may still be assigned to the CPU pool. The standard CPU pool *STDPPOOL and the statistic CPU-Pools *POOL0 and *FOREIGN (SQ-Server) cannot be deleted.

DELETE-VM-CPU-POOL
CPU-POOL-NAME = <name 1..8>

CPU-POOL-NAME = <name 1..8>

Name of the CPU pool.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
	64	VMS4600	CPU pool not yet created
	64	VMS4604	CPU pool still contains a CPU
	64	VMS4614	VM still assigned to CPU pool
	64	VMS4624	VM group still assigned to CPU pool

Guaranteed messages

VMS4605

Example

```
/delete-vm-cpu-pool cpu-pool-name=pcust1
VMS4605 CPU POOL 'PCUST1' DELETED
```

See also the detailed [“Example of managing CPU pools” on page 233](#).

DELETE-VM-GROUP

Delete VM group (S servers)

[SQ] No VM groups are available on SQ servers. The command is rejected with message VMS3011 and with general return code VMS9004.

/DELETE-VM-GROUP is used to delete a VM group.

This command is only permitted for the VM2000 administrator.

VMS4405, VMS4629

DELETE-VM-GROUP
VM-GROUP-NAME = <name 1..8>

VM-GROUP-NAME = <name 1..8>

Name of the VM group.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
	64	VMS4400	VM group not yet created
	64	VMS4404	VM group still contains VMs

Guaranteed messages

VMS4405, VMS4629

Example

```
/delete-vm-group vm-group-name=gcust1
VMS4629 VM GROUP 'GCUST1' REMOVED FROM CPU POOL '*STDPPOOL'
VMS4405 VM GROUP 'GCUST1' DELETED
```

See also the detailed [“Example of operating VM groups”](#) on page 182.

DETACH-VM-RESOURCES

Detach real CPUs

`/DETACH-VM-RESOURCES` is used to detach attached real normal CPUs and extra CPUs for VM2000 operation.

This command is only permitted for the VM2000 administrator.

DETACH-VM-RESOURCES	
UNIT = *CPU(...) / *EXTRA-CPU(...)	
*CPU(...)	
	CPU-IDENTIFICATION = *ANY(...) / list-poss(63): <x-text 1..2>
	*ANY(...)
	CPU-POOL-NAME = *UNIQUE / *STDPOOL / <name 1..8>
*EXTRA-CPU(...)	
	CPU-IDENTIFICATION = *ALL(...) / *ANY(...) / list-poss(63): <x-text 1..2>
	*ALL(...)
	CPU-POOL-NAME = *UNIQUE / *STDPOOL / <name 1..8> / *ALL
	*ANY(...)
	CPU-POOL-NAME = *UNIQUE / *STDPOOL / <name 1..8>

UNIT = *CPU(...)

Specifies real normal CPUs to be detached.



Detaching a real CPU has no effect on the reconfiguration state (attached/detached) of the virtual CPUs of a VM. Virtual CPUs may be detached in the BS2000 guest system by means of the BS2000 command `/DETACH-DEVICE UNIT=*CPU(CPU-IDENTIFIER=<mn>)`.

The last available real normal CPU cannot be detached.



The last available real CPU of a CPU pool in which VMs are still active and the last available real normal CPU cannot be detached.

CPU-IDENTIFICATION = *ANY(...)

Selects any real normal CPU from a CPU pool.

CPU-POOL-NAME =

Determines the CPU pool.

CPU-POOL-NAME = *UNIQUE

A real normal CPU from the CPU pool (*STDPOOL) is selected (operand provided for reasons of compatibility with earlier VM2000 versions).

CPU-POOL-NAME = *STDPOOL / <name 1..8>

Name of the CPU pool.

CPU-IDENTIFICATION = list-poss(63): <x-text 1..2>

Identifier of the normal CPU.

UNIT = *EXTRA-CPU(...)

Specifies the extra CPUs to be detached, see [page 222](#).

CPU-IDENTIFICATION = *ALL(...)

Selects all extra CPUs from a CPU pool.

CPU-POOL-NAME =

Determines the CPU pool.

CPU-POOL-NAME = *UNIQUE

All extra CPUs from the CPU pool (*STDPOOL) are selected (operand provided for reasons of compatibility with earlier VM2000 versions).

CPU-POOL-NAME = *STDPOOL / <name 1..8>

Name of the CPU pool.

CPU-POOL-NAME = *ALL

All extra CPUs from all CPU pools are selected.

CPU-IDENTIFICATION = *ANY(...)

Selects any extra CPU from a CPU pool.

CPU-POOL-NAME =

Determines the CPU pool.

CPU-POOL-NAME = *UNIQUE

One extra CPU from the CPU pool *STDPOOL is selected (operand provided for reasons of compatibility with earlier VM2000 versions).

CPU-POOL-NAME = *STDPOOL / <name 1..8>

Name of the CPU pool.

CPU-IDENTIFICATION = list-poss(63): <x-text 1..2>

Identifier of the extra CPU.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
1	0	CMD0001	No action required
	64	VMS3002	Invalid operand value (operand CPU-ID)
	64	VMS4043	CPU not known in system
	64	VMS4048	CPU is last available CPU. Reconfiguration not possible
	64	VMS4078	CPU not detached due to CPU attribute
	64	VMS4079	No adequate CPU attached
	64	VMS4151	CPU could not be detached due to its state
	64	VMS4270	No extra CPU in the configuration
	64	VMS4600	CPU pool not yet created
	64	VMS4609	Several CPU pools created
	64	VMS4611	Selected CPUs do not belong to CPU pool
	64	VMS4617	Last attached CPU of an active CPU pool
	64	VMS9005	Command not or only partially processed

Guaranteed messages

VMS4043, VMS4045, VMS4047, VMS4048, VMS4077, VMS4078, VMS4151

Example

```

/show-vm-resources information=*cpu _____ (1)
VM-ID      CPU-POOL GR-CPU-Q CPU-Q/ EFF-Q MAX-U/ EFF-U PROC
 1 MONITOR *STDPOOL      14.00/ 50.00 100.00/ 50.00  2

VM-ID      SCHED ACT-I VIRT CPUS
 1 MONITOR  TS     NO   00(RUN),01(RUN)

CPU-ATTR   CPU-POOL      REAL CPUS
NORMAL     *STDPOOL      0A(ATT),0B(ATT),0C(ATT),0D(ATT)
EXTRA      *STDPOOL      0E(ATT),0F(ATT) _____ (2)
SPARE      NONE
FOREIGN    *POOL0        00(ATT),01(ATT),02(ATT),03(ATT)
           *FOREIGN    04(ATT),05(ATT),06(ATT),07(ATT),
           08(ATT),09(ATT)

/detach-vm-resources unit=*extra-cpu _____ (3)
VMS4077 EXTRA-CPU OE DETACHED
VMS4077 EXTRA-CPU OF DETACHED

/detach-vm-resources unit=*cpu(cpu-id=0d) _____ (4)
VMS4045 CPU 0D DETACHED

```

```

/show-vm-resources information=*cpu _____ (5)
VM-ID      CPU-POOL  GR-CPU-Q  CPU-Q/ EFF-Q  MAX-U/ EFF-U  PROC
1 MONITOR  *STDPOOL                14.00/ 50.00 100.00/ 50.00  2

VM-ID      SCHED ACT-I  VIRT CPUS
1 MONITOR  TS      NO      00(RUN),01(RUN)

CPU-ATTR   CPU-POOL      REAL CPUS
NORMAL     *STDPOOL      0A(ATT),0B(ATT),0C(ATT),0D(DET)
EXTRA      *STDPOOL      0E(DET),0F(DET) _____ (6)
SPARE
FOREIGN    *POOL0       00(ATT),01(ATT),02(ATT),03(ATT)
           *FOREIGN  04(ATT),05(ATT),06(ATT),07(ATT),
           08(ATT),09(ATT)

```

- (1) The VM2000 administrator requests information on the CPU state.
- (2) All real normal CPUs and the extra CPU are attached.
- (3) All extra CPUs are detached.
- (4) The real normal CPU 0D and the extra CPUs are detached.
- (5) The VM2000 administrator requests information on the CPU state.
- (6) The normal CPU 0D is attached.

See also the detailed [“Example of managing CPU pools” on page 233](#).

END-VM-DIALOG

Terminate administration or operation

`/END-VM-DIALOG` is used to terminate an ADMIN dialog or VC dialog opened with a corresponding `/BEGIN-VM-DIALOG`. Termination of guest system operation with BS2000 consoles is described on [page 100](#).

`/END-VM-DIALOG` also clears the connection to `$VMCONS` if the terminal was only connected to `$VMCONS` (`OPNCON`), but no dialog has yet been opened.

When the `/DELETE-VM` command is issued, all ADMIN dialogs with the VM and all VC dialogs with the guest system on the VM are terminated implicitly.



The VM administrator can enter `/END-VM-DIALOG` for his/her VM.

If the VM2000 administrator enters `/END-VM-DIALOG` for a VM other than the monitor VM, all affected dialogs of the VM are terminated, depending on the specification in the `TYPE` operand (ignoring the `DISCONNECT` operand).

If the VM2000 administrator enters `/END-VM-DIALOG` for the monitor VM from a BS2000 console, all ADMIN dialogs of the monitor VM are terminated.

The output of periodic monitored data (`/SHOW-VM-STATUS PERIOD=<integer>`) is terminated if the associated ADMIN dialog is terminated.

If VC dialogs only are conducted across a `$VMCONS` connection, `/END-VM-DIALOG` must be specified with the prefix `VM00:` to distinguish it from inputs at the guest system.

END-VM-DIALOG

```
VM-IDENTIFICATION = *CURRENT / <integer 1..99> / <name 1..8>
,DISCONNECT = *YES / *NO
,TYPE = *BOTH / *ADMIN / *VC
```

VM-IDENTIFICATION =

Identification of the virtual machine.

VM-IDENTIFICATION = *CURRENT

The VM identification assigned previously by means of `/BEGIN-VM-DIALOG` is used.

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>

Identifies the virtual machine by means of a VM index or VM name.

DISCONNECT = *YES / *NO

Determines whether or not the connection to \$VMCONS is to be cleared down. The DISCONNECT operand is only evaluated when the last dialog of a connection to \$VMCONS is terminated. This operand is not evaluated for ADMIN dialogs from a privileged user task. It is also evaluated if the VM2000 administrator terminates dialogs of other VMs.

TYPE =

Defines the type of dialog to be terminated for command input via \$VMCONS. This operand is ignored with command input from a privileged user task, as in this case only one ADMIN dialog can be conducted.

TYPE = *BOTH

Both the ADMIN dialog and all the VC dialogs are to be terminated.

TYPE = *ADMIN

The ADMIN dialog is to be terminated.

TYPE = *VC

The VC dialogs are to be terminated.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
	64	VMS1037	No corresponding dialog has been opened
	64	VMS1039	TYPE specification is not allowed
	64	VMS4000	VM not yet created

Example (using \$VMCONS)

```

vm04:/end-vm-dialog_____ (1)
VM01:VMS1012 PARTNER HAS TERMINATED AN ADMINISTRATION DIALOG WITH
    VIRTUAL MACHINE (4,SYSTEM4). ADMINISTRATION-ID USED '@003' _____ (2)
VM01:VMS1032 PARTNER HAS TERMINATED A VC DIALOG WITH
    VIRTUAL MACHINE (4,SYSTEM4)
VM04:VMS3000 VM2000 COMMAND /END-VM-DIALOG FOR
    VIRTUAL MACHINE (4,SYSTEM4) PROCESSED SUCCESSFULLY
VM04:NBRO740 COMMAND COMPLETED 'END-VM-DIALOG';
    (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
@: OMS0006 CONNECTION TO PARTNER ($VMCONS/SHIRKVM4) LOST
    (PAC=VM04; CODE=00000000)

```

- (1) The VM administrator terminates the ADMIN dialog and the VC dialog.
- (2) The VM2000 administrator is informed of the termination of the ADMIN dialog and the VC dialog.

EXTEND-VM-MEMORY

Increase main memory for a VM

`/EXTEND-VM-MEMORY` enables you to enlarge the main memory of a VM, even when the guest system is running. A new value can be set for the minimum main memory size. The reconfiguration of the main memory can be monitored with a monitor job variable.

This command is only permitted for the VM2000 administrator.

SQ On SQ servers the main memory can be extended to the maximum size set, see [page 141](#) and [page 271](#).

On SQ servers the main memory of a VM cannot be enlarged while it is being migrated (Live Migration).

Main memory reconfiguration is possible for VMs in the INIT-ONLY or DOWN states. In the case of active guest systems (VM state RUNNING), main memory reconfiguration is only possible after `SYSTEM READY` for the guest system. Main memory reconfiguration is not possible for VMs in the IN HOLD (WAIT) status.

Since dynamic reconfiguration of main memory requires measures to be taken both in the (active) guest system and in VM2000, the process can be time-consuming. The actual main memory reconfiguration takes place asynchronously after the command has been accepted. During this time any further `/EXTEND-VM-MEMORY` or `/REDUCE-VM-MEMORY` command for the same VM will be rejected.

The `/MOVE-VM` (S servers) and `/START-VM` commands can be executed during main memory reconfiguration. In this case, the new main memory size is used as the main memory size for the relevant VM. The same applies when the guest system is restarted.

EXTEND-VM-MEMORY

```

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>
,NEW-MEMORY-SIZE = *UNCHANGED / *MAXIMUM /
                  <integer 1..1048576>(DIMENSION = *MEGABYTE / *GIGABYTE)
,NEW-MIN-MEMORY-SIZE = *UNCHANGED / *MAXIMUM /
                      <integer 1..1048576>(DIMENSION = *MEGABYTE / *GIGABYTE)
,MONJV = *NONE / <filename 1..54 without-gen-vers>
,JV-PASSWORD = *NONE / <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>

```

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>

Identifies the virtual machine by means of a VM index or VM name. The monitor VM's main memory can also be enlarged.

**NEW-MEMORY-SIZE = *UNCHANGED / *MAXIMUM / <integer 1..1048576>
(DIMENSION = *MEGABYTE / *GIGABYTE)**

New size of main memory after enlargement for the relevant VM up to a size of 1 Tbyte (terabyte; 1 Tbyte = 1024 Gbytes = 1 048 576 Mbytes).

For a guest system BS2000/OSD V6.0 the maximum value is 32767 Mbytes.

- [S] If *MAXIMUM is specified, the main memory size is increased up to the next VM or to the end of the available main memory. For a guest system BS2000/OSD V6.0 the maximum value is 32767 Mbytes.
- [SQ] The value must be a multiple of 2 Mbytes. It may not be greater than the MAX-MEMORY-SIZE of the VM. When *MAXIMUM is specified, the main memory size is extended to the MAX-MEMORY-SIZE of the VM. When the NEW-MEMORY-SIZE of the VM is greater than the main memory which is available, the VM is assigned the main memory which is available.

**NEW-MIN-MEMORY-SIZE = *UNCHANGED / *MAXIMUM / <integer 1..1048576>
(DIMENSION = *MEGABYTE / *GIGABYTE)**

New minimum size of the main memory for the relevant VM when main memory is reconfigured. The new minimum size of the main memory must not be less than the previous minimum size.

For a guest system BS2000/OSD V6.0 the maximum value is 32767 Mbytes.

- [SQ] The value must be a multiple of 2 Mbytes.

If *MAXIMUM is specified, the main memory size defined by (NEW-)MEMORY-SIZE is chosen for the VM. For a guest system BS2000/OSD V6.0 the maximum value is 32767 Mbytes.

MONJV =

Defines a MONJV for monitoring the reconfiguration of the main memory (see [page 272](#)).

MONJV = *NONE

No MONJV is defined.

MONJV = <filename 1..54 without-gen-vers>

The specified MONJV is supplied with data by VM2000; the sender of the command must therefore have write permission for the specified MONJV. If no catalog ID or user ID is specified in the file name, VM2000 completes the file name, in accordance with the runtime environment of the sender of the command. If a file name which does not yet have a job variable is specified, VM2000 creates the MONJV as shareable and without a password.

JV-PASSWORD = *NONE / <c-string 1..4> / <x-string 1..8> / <integer ...>

Password of the MONJV for checking access authorization. The input field is automatically blanked in the guided dialog. The value input is not logged.



This operand is only evaluated if a previously created job variable was specified with MONJV=.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
1	0	CMD0001	No action required
	64	VMS2070	VM in IN HOLD (WAIT) state
	64	VMS4000	VM not yet created
	64	VMS4011	Requested memory size or location not available
	64	VMS4071	Invalid memory size or location
	64	VMS4091	A guest system for which main memory reconfiguration is not possible is running on the VM
	64	VMS4092	Memory reconfiguration is still active
	64	VMS4093	Invalid memory size
	64	VMS4340	Opening of MONJV rejected by subsystem JV
	64	VMS4341	MONJV already assigned to another VM
	128	VMS3042	VM being migrated. Command rejected

Example

```

/show-vm-resources information=*memory _____ (1)
VM-ID      LOC-BEGIN  MEMORY-SIZE
HYPERVISOR 00000000      14 MB
  1 MONITOR 0000000E      128 /    128 MB
  2 SYSTEM2 0000008E      128 /    128 MB
NONE       0000010E      3814 MB
/extend-vm-memory vm-id=2,new-memory-size=256 _____ (2)
/show-vm-resources information=*memory
VMS4094 MEMORY-RECONFIGURATION FOR VIRTUAL MACHINE (2,SYSTEM2) HAS BEEN
        SUCCESSFULLY PROCESSED. CURRENT MEMORY SIZES: 128 / 256 _____ (3)
VM-ID      LOC-BEGIN  MEMORY-SIZE
HYPERVISOR 00000000      14 MB
  1 MONITOR 0000000E      128 /    128 MB
  2 SYSTEM2 0000008E      256 /    128 MB _____ (4)
NONE       0000018E      3686 MB

```

- (1) The VM2000 administrator fetches information on the main memory distribution in VM2000. VM2 has a main memory size of 128 Mbytes and a minimum memory size of 128 Mbytes. Since the guest system on VM2 has already reached SYSTEM READY it is possible to reconfigure the main memory in this guest system. The subsequent memory area has not been assigned, so that the main memory for VM2 can be enlarged.
- (2) The main memory for VM2 is to be enlarged to 256 Mbytes. The minimum main memory size is unchanged.
- (3) After its completion, the result of the asynchronously executing reconfiguration request is reported to the VM2000 administrator as an asynchronous (guaranteed) message in the user event stream or displayed at the next input (main memory reconfiguration is time-consuming).
- (4) The main memory size for VM2 is now 256 Mbytes. This reduces the remaining free main memory.

HOLD-VM

Suspend VM

`/HOLD-VM` is used to suspend one or more VMs. The monitor VM cannot be suspended. After execution of the command, the VM is in the IN HOLD(WAIT) state.

This command can be canceled by entering `/RESUME-VM`. The following rules apply:

- The wait state of a VM suspended by the VM administrator can be canceled by the VM2000 administrator
- The wait state of a VM suspended by the VM2000 administrator cannot be canceled by the VM administrator



Precautions must be taken for guest systems in the MSCF network to ensure that the guest system is not regarded as failed while it is suspended.

Current I/O operations are still executed. However, termination interrupts for these I/O operations are not accepted by the guest system until after the end of the wait state.

Overlapping wait states are described on [page 163](#).



On SQ servers a VM cannot be suspended while it is being migrated (Live Migration).

HOLD-VM
VM-IDENTIFICATION = *CURRENT / *ALL / <integer 1..99> / <name 1..8>

VM-IDENTIFICATION =

Identification of the virtual machine.

VM-IDENTIFICATION = *CURRENT

The VM identification assigned previously by means of `/BEGIN-VM-DIALOG` is used.

VM-IDENTIFICATION = *ALL

All currently executing VMs except the monitor VM are suspended. This operand is only permitted for the VM2000 administrator.

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>

Identifies the virtual machine by means of a VM index or VM name.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
1	0	CMD0001	No action required
	64	VMS1002	Command not permitted for the monitor VM
	64	VMS2101	VM2000 shutdown in progress
	64	VMS4000	VM not yet created
	64	VMS9005	Command not or only partially processed
	128	VMS3042	VM being migrated. Command rejected

Guaranteed messages

VMS2000, VMS2012, VMS3041

Example

```
/show-vm-attributes vm-id=system2 _____ (1)
VM-ID      VM-GROUP(G)  CPU-Q    MAX-U    PROC    MEM-SIZE  MAX-IO
 2 SYSTEM2  *STDPOOL(P) 30.00   100.00   1       1024 MB (100)
```

```
VM-ID      STATE          GUEST   VERSION  ADMIN  PRIV
 2 SYSTEM2  RUNNING        BS2-OSD V09.0A  NO     AS _____ (2)
```

```
/hold-vm vm-id=system2 _____ (3)
VMS2012 VIRTUAL MACHINE (2,SYSTEM2) PLACED IN WAIT STATE BY THE
      VM2000 ADMINISTRATOR
```

```
/show-vm-attributes vm-id=system2
VM-ID      VM-GROUP(G)  CPU-Q    MAX-U    PROC    MEM-SIZE  MAX-IO
 2 SYSTEM2  *STDPOOL(P) 30.00   100.00   1       1024 MB (100)
```

```
VM-ID      STATE          GUEST   VERSION  ADMIN  PRIV
 2 SYSTEM2  WAIT(SEL)      BS2-OSD V09.0A  NO     AS _____ (4)
```

- (1) The VM2000 administrator requests the status of VM2.
- (2) VM2 is in the RUNNING state.
- (3) The VM2000 administrator suspends VM2 by means of /HOLD-VM.
- (4) VM2 is in the state WAIT(SEL).

MODIFY-VM-ATTRIBUTES

Modify attributes of a VM

`/MODIFY-VM-ATTRIBUTES` is used to modify the existing attributes of a VM. The attributes of a VM were set using `/CREATE-VM`.

The following attributes can be modified:

- The CPU quota
- maximum CPU utilization of the VM
- maximum I/O utilization of the VM (S servers)
- password for administration
- command range for VM2000 administrator and VM administrator
- privileges of the VM
- settings for controlling the real CPU (S servers)
- the scope of the trace entries for VM2000 to be logged

This command can only be used by the VM2000 administrator and VM administrators with `ADMIN-PRIVILEGES=*MAXIMUM`.

SQ On SQ servers the attributes of a VM cannot be modified while it is being migrated (Live Migration).

MODIFY-VM-ATTRIBUTES

```

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>
,CPU-QUOTA = *UNCHANGED / <fixed 0.01..99.99> / *BY-VM-GROUP(...)
    *BY-VM-GROUP(...)
        | MEMBER-CPU-QUOTA = *UNCHANGED / <fixed 0.01..99.99>
,MAX-CPU-UTILIZATION = *UNCHANGED / <fixed 0.01..100.00>
,MAX-IO-UTILIZATION = *UNCHANGED / <integer 1..100>
,PASSWORD = *UNCHANGED / *NONE / <c-string 1..8> / <x-string 1..16>
,ADMIN-PRIVILEGES = *UNCHANGED / *STD / *MINIMUM / *MAXIMUM
,PRIVILEGES = *PARAMETERS(...)
    *PARAMETERS(...)
        | IO-RESET = *UNCHANGED / *NO / *YES
        | ,IO-PRIORITY = *UNCHANGED / *NO / *YES
        | ,AUTO-SNAP-ASSIGNMENT = *UNCHANGED / *NO / *YES
        | ,ASSIGN-BY-GUEST = *UNCHANGED / *NO / *YES(...)
            *YES(...)
                | SET-NAME = *STDSET / *ALL / *EXCEPT(...) / *ADD(...) / *REMOVE(...) /
                    list-poss(8): <name 1..8> / *STDSET
                        *EXCEPT(...)
                            | SET-NAME = list-poss(8): <name 1..8> / *STDSET
                        *ADD(...)
                            | SET-NAME = list-poss(8): <name 1..8> / *STDSET
                        *REMOVE(...)
                            | SET-NAME = list-poss(8): <name 1..8> / *STDSET
,VM-ACTIVE-IDLE = *UNCHANGED / *NO / *AT-DEDICATED-CPUS
,TRACE-LEVEL = *UNCHANGED / *OFF / *STD / *MAXIMUM
,VM-NAME = *UNCHANGED / <name 1..8>

```

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>

Identifies the virtual machine by means of a VM index or VM name.

CPU-QUOTA =

This determines whether the CPU quota is to be changed or retained.

CPU-QUOTA = *UNCHANGED

The previous CPU quota or member CPU quota is retained.

CPU-QUOTA = <fixed 0.01..99.99>

New CPU quota for a VM which does not belong to a VM group, see [page 238](#).

CPU-QUOTA = *BY-VM-GROUP(...)

New member CPU quota for a VM which belongs to a VM group (S servers), see [page 176](#).

MEMBER-CPU-QUOTA =

Defines whether the member CPU quota is to be modified or retained.

MEMBER-CPU-QUOTA = *UNCHANGED

The previous member CPU quota is retained

MEMBER-CPU-QUOTA = <fixed 0.01..99.99>

New member CPU quota.

MAX-CPU-UTILIZATION =

Determines whether the maximum CPU utilization of the VM is to be changed or retained.

MAX-CPU-UTILIZATION = *UNCHANGED

The previous maximum CPU utilization of the VM is retained.

MAX-CPU-UTILIZATION = <fixed 0.01..100.00>

New maximum CPU utilization of the VM, see [page 238](#).

If the VM belongs to a VM group, its maximum CPU utilization is also restricted by the “maximum CPU utilization of the VM group” (see [page 176](#)).

MAX-IO-UTILIZATION =

On S servers, determines whether the maximum CPU utilization of the VM is to be changed or retained.

MAX-IO-UTILIZATION = *UNCHANGED

The previous maximum IO utilization of the VM is retained.

MAX-IO-UTILIZATION = <integer 1..100>

New maximum IO utilization of the VM, see [page 360](#).

PASSWORD =

This determines whether the password which determines the right of access in /BEGIN-VM-DIALOG is to be changed or retained. The password can be up to 8 bytes long.

PASSWORD = *UNCHANGED

The previous password is retained.

PASSWORD = *NONE

No password is to be used.

PASSWORD = <c-string 1..8>

New password in character form.

PASSWORD = <x-string 1..16>

New password in hexadecimal form.

ADMIN-PRIVILEGES =

Determines whether the command/function range for VM2000 and VM administrators is to be changed or retained (see [page 192](#)).

ADMIN-PRIVILEGES = *UNCHANGED

The existing command/function range remains valid.

ADMIN-PRIVILEGES = *STD

Standard command/function range for the VM administrator.

ADMIN-PRIVILEGES = *MINIMUM

Restricted command/function range for the VM2000 administrator.



For security reasons, a command/function range restricted by means of *MINIMUM **cannot** be revoked during the session.

ADMIN-PRIVILEGES = *MAXIMUM

Extended command/function range for the VM administrator.

PRIVILEGES = *PARAMETERS(...)

Describes privileges of the VM. If *UNCHANGED is specified, the privileges of the VM remain unchanged.

IO-RESET = *UNCHANGED / *NO / *YES

Specifies on S servers whether the channels are reset **in the hardware** for this VM at /START-VM or when the guest system is restarted or the channels reconfigured (see [page 145](#)).

IO-PRIORITY = *UNCHANGED / *NO / *YES

Specifies on S servers whether a waiting virtual CPU of the VM is executed again on a real CPU by the VM2000 hypervisor immediately after the end of pending input/output for this VM (see [page 146](#)).



The total number of virtual CPUs of all the VMs with this privilege must not be greater than that of the real normal CPUs of the server (excluding spare and extra CPUs).

AUTO-SNAP-ASSIGNMENT = *UNCHANGED / *YES / *NO

Determines whether the guest system of the VM may implicitly assign snap units for Snapsets without the VM and device being assigned the `ASSIGN-BY-GUEST` privilege or attribute.

ASSIGN-BY-GUEST = *UNCHANGED / *NO / *YES(...)

Specifies whether devices can also be assigned to this VM implicitly (*YES). If not, (*NO) the devices have to be assigned to the VM explicitly, see [page 146](#).

SET-NAME =

Determines the assignment sets from which the guest system of the VM may implicitly assign itself devices. A VM can be explicitly assigned up to 28 assignment sets by means of explicit list specification.

SET-NAME = *STDSET / list-poss (8): <name 1..8> / *STDSET

Explicit specification of the assignment sets.

SET-NAME = *ALL

All existing and future assignment sets.

SET-NAME = *EXCEPT(SET-NAME = list-poss (8): <name 1..8> / *STDSET)

All existing and future assignment sets with the exception of those which are specified with `SET-NAME`.

SET-NAME = *ADD(SET-NAME = list-poss (8): <name 1..8> / *STDSET)

Extends the existing set of assignment sets for the VM.

SET-NAME = *REMOVE(SET-NAME = list-poss (8): <name 1..8> / *STDSET)

Reduces the existing set of assignment sets for the VM.

VM-ACTIVE-IDLE =

Determines on S servers whether, in the event of fixed CPU assignment (dedicated CPUs), a VM still retains control over a real CPU if the VM's virtual CPU which runs on this is inactive (interruptible wait state, "Idle"), see [page 147](#).

VM-ACTIVE-IDLE = *UNCHANGED

The value set is retained.

VM-ACTIVE-IDLE = *NO

The VM2000 hypervisor deletes the real CPU if the VM's virtual CPU which runs on this is inactive (interruptible wait state).

VM-ACTIVE-IDLE = *AT-DEDICATED-CPUS

The VM retains control over the real CPU even if the VM's virtual CPU which runs on this is inactive (interruptible wait state, "Idle").

This idle time is then indicated in the VM2000 accounting records, with /SHOW-VM-STATUS (VM-ACTIVE output column) and in the VM2000 report of openSM2 as a time in which the VM actively uses the real CPU.



With fixed CPU assignment, VM-ACTIVE-IDLE=*AT-DEDICATED-CPUS is only effective if the VM's maximum CPU utilization (see [page 142](#)) is not restricted.

TRACE-LEVEL =

Specifies the scope of the trace entries for VM2000 to be logged (see [page 409](#)). This operand can only be specified for the monitor VM.

TRACE-LEVEL = *UNCHANGED

The current TRACE scope remains unchanged. The default value after /CREATE-VM is *OFF.

TRACE-LEVEL = *OFF

Only entries in the error trace are logged (system errors).

TRACE-LEVEL = *STD

Entries in the error trace and the important points of the runtime trace and the \$VMCONS trace are logged.

TRACE-LEVEL = *MAXIMUM

All trace entries are logged.

VM-NAME =

Determines whether accounting records are to be written.

VM-NAME = *UNCHANGED

The previous VM name is retained.

VM-NAME = <name 1..8>

The name of a VM can no longer be changed. The old name must be specified. The entry VM-NAME=<old-name> causes the accounting records to be written (see [page 381](#)).

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
2	0	CMD0001	Decimal specifications rounded to 2 decimal places
	64	VMS3002	Invalid operand value
	64	VMS3033	ADMIN-PRIVILEGES=*MAX not permissible for monitor VM
	64	VMS3034	ADMIN-PRIVILEGES=*MIN only permissible for monitor VM
	64	VMS3035	IO-PRIORITY cannot be set (number of virtual CPUs with a privilege too high)
	64	VMS4000	VM not yet created
	64	VMS4200	Assignment set not yet created
	64	VMS4211	Maximum number of assignment sets already assigned to the VM
	64	VMS4428	VM does not belong to a VM group
	64	VMS4450	CPU quota must be specified via MEMBER-CPU-QUOTA
	64	VMS9004	Functionality not supported on current HSI or with current parameter assignment
	128	VMS3042	VM being migrated. Command rejected

Example

```
/modify-vm-attributes vm-id=system2,cpu-quota=20,max-cpu-utilization=25,
  privileges=*parameters(assign-by-guest=*yes(set-name=*stdset)) _____ (1)
/show-vm-attributes vm-id=system2,information=*std _____ (2)
```

```
      CPU-POOL(P)
VM-ID   VM-GROUP(G)  CPU-Q   MAX-U   PROC   MEM-SIZE  MAX-IO
  2 SYSTEM2  *STDPOOL(P)  20.00  25.00    1    1024 MB (100)
```

```
VM-ID   STATE           GUEST  VERSION  ADMIN  PRIV
  2 SYSTEM2  INIT-ONLY      UNKNOWN          NO    AG,AS
```

```
/show-vm-attributes vm-id=system2,information=*privileges
```

```
VM-ID   PRIV  DETAILS
  2 SYSTEM2  AG    *STDSET
      AS
```

- (1) The VM2000 administrator changes the attributes of VM2. The new CPU quota is 20. The maximum CPU utilization is 25. The ASSIGN-BY-GUEST for the assignment set *STDSET privilege is assigned to VM2.
- (2) The VM administrator queries the attributes and the details of the privileges for VM2.


See also the detailed [“Example of operating VM groups” on page 182.](#)

MODIFY-VM-DEVICE-ATTRIBUTES

Modify device attributes


`/MODIFY-VM-DEVICE-ATTRIBUTES` is used to specify whether the specified devices can also be assigned implicitly to a VM by a guest system.


For this purpose devices have the device attribute `ASSIGN-BY-GUEST`. The default setting is `ASSIGN-BY-GUEST=*NO`, i.e. the devices can initially not be assigned implicitly by a VM's guest system.

-  The default setting for this device attribute can be modified for S servers in the VM2000 parameter file (see [page 54](#)).

The default setting for this device attribute can be modified for selected devices or for all existing devices using `/MODIFY-VM-DEVICE-ATTRIBUTES UNITS=...` and for all new devices using `/MODIFY-VM-DEVICE-ATTRIBUTES UNITS=*NEW-DEVICES`. New devices (dynamic I/O configuration modification, see [page 361](#)) then assume the new setting as a device attribute.

DUMMY tape devices should not be assigned implicitly, see [page 336](#).

-  Virtual consoles cannot be assigned implicitly by a guest system. Alias devices (PAV) can only be assigned implicitly (the attribute is ignored for these devices).

-  The usage mode of a device does not change with this command (see `/MODIFY-VM-DEVICE-USAGE` on [page 495](#)).

A message is only issued for a device in the event of an error.

This command is only permitted for the VM2000 administrator.

MODIFY-VM-DEVICE-ATTRIBUTES

```

UNITS = *ALL / *NEW-DEVICES / *PUBSET-DEVICES(...) / *BY-ASSIGNMENT-SET(...) /
    list-poss(28): *DEVICE-RANGE(...) / <alphanum-name 1..4 with-wild(20)>

*PUBSET-DEVICES(...)
    |
    |   PUBSET = *BY-PUBRES-DEVICE(...) / <cat-id 1..4>
    |   |
    |   |   *BY-PUBRES-DEVICE(...)
    |   |   |
    |   |   |   UNIT = <alphanum-name 2..2> / <x-text 4..4>
    |
*BY-ASSIGNMENT-SET(...)
    |
    |   SET-NAME = *STDSET / <name 1..8>
*DEVICE-RANGE(...)
    |
    |   FROM = <alphanum-name 2..2> / <x-text 4..4>
    |   ,TO = <alphanum-name 2..2> / <x-text 4..4>
,ASSIGN-BY-GUEST = *NO / *YES(...)
*YES(...)
    |
    |   SET-NAME = *STDSET / <name 1..8>

```

UNITS =

Specifies the devices to be assigned the attribute.



The setting for new devices can only be modified using `UNITS=*NEW-DEVICES`.

UNITS = *ALL

The attribute is to be assigned to all existing devices.

UNITS = *NEW-DEVICES

The attribute is to be assigned to all new devices (dynamic I/O configuration modification).

UNITS = *PUBSET-DEVICES(...)

The attribute is to be modified for all the devices of a pubset.



See also [“Notes on pubset specification in the VM2000 commands” on page 313](#) and [“Pubset reconfiguration” on page 315](#).

PUBSET = *BY-PUBRES-DEVICE(...)

Determines the devices of the pubset by means of the pubres device.

UNIT = <alphanum-name 2..2> / **<x-text 4..4>**

Mnemonic device name of the pubres device.

PUBSET = <cat-id 1..4>

Catalog ID of the pubset. The pubset must be entered together with its pubres device in the monitor system's MRS catalog.

UNITS = *BY-ASSIGNMENT-SET(...)

The attribute is to be modified for all devices of an assignment set.

SET-NAME = *STDSET / <name 1..8>

Determines the devices of the assignment set by means of the set name.

UNITS = *DEVICE-RANGE(...)

All the devices in this name range are to be assigned the attribute.



Device names <x-text 4..4> (in *DEVICE-RANGE) are selected from FROM to TO in hexadecimal order. Otherwise, device names <alphanum-name 2..2> are selected in alphabetical order.

FROM = <alphanum-name 2..2> / <x-text 4..4>

Defines the start of the name range.

TO = <alphanum-name 2..2> / <x-text 4..4>

Defines the end of the name range. The mnemonic device name must be in the same value range (2- or 4-digit mn) as the FROM device name and must be greater than or equal to the FROM device name in ascending order in the value range.

UNITS = <alphanum-name 1..4 with-wild(20)>

Specifies the devices to be assigned an attribute.

ASSIGN-BY-GUEST = *NO / *YES(...)

Specifies whether the devices can also be assigned to a VM by a guest system implicitly. See [page 146](#).

ASSIGN-BY-GUEST = *NO

Device assignment can only be executed using the VM2000 commands /ADD-VM-DEVICE or /SWITCH-VM-DEVICES. *Exception:* Alias devices (see [page 347](#)) and snap units for Snapsets (see [page 336](#)).

ASSIGN-BY-GUEST = *YES(...)

Device assignment can also be executed implicitly by the guest system.

SET-NAME = *STDSET / <name 1..8>

The selected devices are assigned to the specified assignment set.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
1	0	VMS4180	Specified set of devices empty
2	0	VMS4203	No devices are assigned to the assignment set
2	0	VMS4241	Attribute not changed for all devices of the pubset
2	0	VMS4242	Attribute changed only for the pubres device of the pubset
	64	VMS3002	Invalid operand value (e.g. *DEVICE-RANGE operand)
	64	VMS4006	Device not available
	64	VMS4171	ASSIGN-BY-GUEST is not possible for virtual consoles
	64	VMS4200	Assignment set not yet created
	64	VMS4223	Command not executed (operand *PUBSET-DEVICES)
	64	VMS9005	Command could not be processed or could only be partially processed

Guaranteed messages

VMS4171

Only if *PUBSET-DEVICES is specified: VMS4223, VMS4240, VMS4241, VMS4242

Only if UNITS is specified without *DEVICE-RANGE and without wildcards: VMS4006

Example

```
/modify-vm-device-attributes
    units=*pubset-devices(pubset=*by-pubres-device(unit=d4)),
    assign-by-guest=*yes(set-name=sys2impl) _____ (1)
VMS4240 DEVICE ATTRIBUTES OF PUBSET 'VM04' WITH PUBRES DEVICE 'D4' MODIFIED
```

- (1) The VM2000 administrator specifies the `ASSIGN-BY-GUEST` attribute for the pubset with the pubres device `D4` and assigns the pubset devices to the assignment set `SYS2IMPL`.

See also the detailed examples for device and pubset management on [page 294](#) and on [page 324](#).

MODIFY-VM-DEVICE-USAGE

Modify usage mode of the devices of a VM

`/MODIFY-VM-DEVICE-USAGE` changes the usage mode of VM devices, see [page 292](#).

Individual devices are addressed using their mnemonic device names. The devices of a pubset are determined by means of the catalog ID of the pubset or the mnemonic device name of the pubres device.

This command can only be used by the VM2000 administrator and VM administrators with `ADMIN-PRIVILEGES=*MAXIMUM`.

The `SHARED` usage mode can be set for exclusively assigned devices if the device permits this (disk devices, for example, with the exception of the IPL device of the VM).

The `EXCLUSIVE` usage mode can be set for shared devices which are assigned to only one VM at the time of command execution (`SHARED` usage mode with direct I/O, `SH(D)`).



This command does not change the attributes of the device (see `/MODIFY-VM-DEVICE-ATTRIBUTES` on [page 491](#)).



Alias devices always have the same usage mode as the associated base device. The usage mode of a base device can be changed using this command (the usage mode of its alias devices thus changes). The command cannot be used explicitly for an alias device.

MODIFY-VM-DEVICE-USAGE

```

UNITS = *PUBSET-DEVICES(...) /
          list-poss(28): *DEVICE-RANGE(...) / <alphanum-name 1..4 with-wild(20)>
*PUBSET-DEVICES(...)
  |   PUBSET = *BY-PUBRES-DEVICE(...) / <cat-id 1..4>
  |   *BY-PUBRES-DEVICE(...)
  |   |   UNIT = <alphanum-name 2..2> / <x-text 4..4>
*DEVICE-RANGE(...)
  |   FROM = <alphanum-name 2..2> / <x-text 4..4>
  |   ,TO = <alphanum-name 2..2> / <x-text 4..4>
,VM-IDENTIFICATION = <integer 1..99> / <name 1..8>
,USAGE = *EXCLUSIVE / *SHARED

```

UNITS =

Specifies the VM devices whose usage mode is to be changed.

UNITS = *PUBSET-DEVICES(...)

The usage mode is to be modified for all devices of a pubset.



See also [“Notes on pubset specification in the VM2000 commands”](#) on page 313.

PUBSET = *BY-PUBRES-DEVICE(...)

Determines the devices of the pubset by means of the pubres device.

UNIT = <alphanum-name 2..2> / <x-text 4..4>

Mnemonic device name of the pubres device.

PUBSET = <cat-id 1..4>

Catalog ID of the pubset. The pubset must be entered together with its pubres device in the monitor system's MRS catalog.

UNITS = *DEVICE-RANGE(...)

Defines the start of the name range.



Device names `<x-text 4..4>` (in `*DEVICE-RANGE`) are selected from FROM to TO in hexadecimal order. Otherwise, device names `<alphanum-name 2..2>` are selected in alphabetical order.

If UNITS is specified with `*DEVICE-RANGE` or with wildcards, alias devices and devices that are not assigned to the VM are ignored.

FROM = <alphanum-name 2..2> / <x-text 4..4>

Defines the start of the name range.

TO = <alphanum-name 2..2> / <x-text 4..4>

Defines the end of the name range. The mnemonic device name must be in the same value range (2- or 4-digit mn) as the FROM device name and must be greater than or equal to the FROM device name in ascending order in the value range.

UNITS = <alphanum-name 1..4 with-wild(20)>

Specifies the devices whose usage mode is to be changed.

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>

Identifies the virtual machine by means of a VM index or VM name.

USAGE =

Specifies the new usage mode.

USAGE = *EXCLUSIVE

The device is assigned to the VM exclusively. Exclusively assigned devices are listed in the outputs of the VM2000 information commands under the keyword `EXCL`.

USAGE = *SHARED

The device is declared as shareable (`SHARED`), see also [page 292](#). Shared devices are listed in the outputs of the VM2000 information commands under the keyword `SHARED`.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
1	0	CMD0001	No action required
1	0	VMS4035	No corresponding device found (perhaps only alias devices were specified)
1	0	VMS4180	Specified set of devices empty
2	0	VMS4231	Usage mode not changed f or all devices of the pubset
	64	VMS3002	Invalid operand value (e.g. *DEVICE-RANGE operand)
	64	VMS4000	VM not yet created
	64	VMS4006	Device not available
	64	VMS4007	Device is not assigned to the virtual machine
	64	VMS4032	Device cannot be assigned to several VMs at the same time
	64	VMS4172	Device is assigned to several VMs
	64	VMS4190	Device is an alias device (PAV)
	64	VMS4223	Command not executed (operand *PUBSET-DEVICES)
	64	VMS9005	Command not or only partially processed

Guaranteed messages

VMS4032, VMS4172

Only if *PUBSET-DEVICES is specified: VMS4223, VMS4230, VMS4231

Only if UNITS is specified without *DEVICE-RANGE and without wildcards: VMS4006, VMS4007, VMS4190

Example

```
/modify-vm-device-usage vm-id=system2,usage=*shared  
    units=*pubset-devices(pubset=*by-pubres-device(unit=d4)) _____ (1)  
VMS4230 DEVICE USAGE OF PUBSET 'VM04' WITH PUBRES DEVICE 'D4' MODIFIED
```

- (1) The VM2000 administrator wants to set the usage mode `SHARED` for the devices of the pubset with the pubres device `D4`.

See also the detailed examples for device and pubset management on [page 294](#) and on [page 324](#).

MODIFY-VM-GROUP-ATTRIBUTES

Modify attributes of a VM group (S servers)

SQ No VM groups are available on SQ servers. The command is rejected with message VMS3011 and with general return code VMS9004.

/MODIFY-VM-GROUP-ATTRIBUTES is used to modify the attributes specified for a VM group. These attributes were initially defined with /CREATE-VM-GROUP. The following changes can be made:

- CPU quota of the VM group
- Maximum CPU utilization of the VM group

This command is only permitted for the VM2000 administrator.

MODIFY-VM-GROUP-ATTRIBUTES
VM-GROUP-NAME = <name 1..8> ,CPU-QUOTA = <u>*UNCHANGED</u> / <fixed 0.01..99.99> ,MAX-CPU-UTILIZATION = <u>*UNCHANGED</u> / <fixed 0.01..100.00>

VM-GROUP-NAME = <name 1..8>

Name of the VM group.

CPU-QUOTA = *UNCHANGED / <fixed 0.01..99.99>

CPU quota of the VM group, see [page 176](#).

MAX-CPU-UTILIZATION = *UNCHANGED / <fixed 0.01..100.00>

Maximum CPU utilization of the VM group, see [page 176](#).

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
1	0	CMD0001	No action required
2	0	CMD0001	Decimal specifications rounded to 2 decimal places
	64	VMS4400	VM group not yet created

Example

```
/modify-vm-group-attributes vm-group-name=gcust1,cpu-quota=30,  
                             max-cpu-utilization=30
```

See also the detailed [“Example of operating VM groups”](#) on page 182.

MODIFY-VMCONS-PARAMETERS

Modify parameters of \$VMCONS

The \$VMCONS parameters are set in the parameter file of the VM2-MON subsystem. /MODIFY-VMCONS-PARAMETERS is used to modify the following parameters

- the OPERID password for the UCON logon of \$VMCONS
- the password for establishing the connection to \$VMCONS

This command is only permitted for the VM2000 administrator.

The new passwords are valid for new connections after the command has been executed. Existing connections continue to run unchanged.

The new passwords remain valid until the end of the current session or until the command is entered again. They also remain valid after a restart of \$VMCONS. However, they are no longer available after an automatic restart of the monitor system. Changes for the next session must be made in the parameter file of the VM2-MON subsystem.

MODIFY-VMCONS-PARAMETERS

```
OPERID-PASSWORD = *UNCHANGED / *NONE / <c-string 1..8>
,PASSWORD =*UNCHANGED / *NONE / <c-string 1..4> / <x-string 1..8>
```

OPERID-PASSWORD =

This operand informs \$VMCONS that the password for the assigned OPERID has been changed. The password itself is changed by means of the BS2000 command /MODIFY-USER in the monitor system. The password can be up to 8 characters long.

OPERID-PASSWORD = *UNCHANGED

The previous password is retained.

OPERID-PASSWORD = *NONE

No password is to be defined.

OPERID-PASSWORD = <c-string 1..8>

New password in character form.

PASSWORD =

Specifies whether the password defined for establishing the connection to \$VMCONS is to be modified or retained. The password can be up to 4 bytes long.

PASSWORD = *UNCHANGED

The previous password is retained.

PASSWORD = *NONE

No password is to be defined.

PASSWORD = <c-string 1..4>

New password in character form.

PASSWORD = <x-string 1..8>

New password in hexadecimal form.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed

MOVE-VM**Move a VM within main memory (S servers)**

SQ A VM need not be moved on SQ servers. The command is rejected with message VMS3011 and with general return code VMS9004.

/MOVE-VM serves to move a VM in VM2000 main memory. Overlapping with the old VM main memory is possible. The monitor VM cannot be moved.

This command is only permitted for the VM2000 administrator.

MOVE-VM
<p>LOCATION = *BEFORE-VM(...) / *AFTER-VM(...) / *MEMORY-END / <x-text 1..8>(…)</p> <p> *BEFORE-VM(...)</p> <p> VM-IDENTIFICATION = *NEXT / <integer 2..99> / <name 1..8></p> <p> *AFTER-VM(...)</p> <p> VM-IDENTIFICATION = *PREVIOUS / <integer 1..99> / <name 1..8></p> <p> <x-text 1..8>(…)</p> <p> DIMENSION = *BYTE / *MEGABYTE</p> <p>,SIDE-BOUNDARIES = *CHECK / *IGNORE ¹</p> <p>,VM-IDENTIFICATION = <integer 2..99> / <name 1..8></p> <p>,MSCF-COORDINATED = *IF-POSSIBLE / *YES / *NO</p>

¹ Operand no longer available; it is only specified for compatibility

LOCATION =

Specifies the new location of the VM in VM2000 main memory.

LOCATION = *BEFORE-VM(VM-IDENTIFICATION = *NEXT / <integer 2..99> / <name 1..8>)

Specifies the VM before which the new VM is to be located. If *NEXT is specified, the VM is located before the next VM.

LOCATION = *AFTER-VM(VM-IDENTIFICATION = *PREVIOUS / <integer 1..99> / <name 1..8>)

Specifies the VM after which the new VM is to be located. If *PREVIOUS is specified, the VM is located after the previous VM.

LOCATION = *MEMORY-END

The VM is to be located at the end of the available main memory.

LOCATION = <x-text 1..8>(DIMENSION = *BYTE / *MEGABYTE)

Specifies the address with the dimension. The address must be a multiple of 1 Mbyte. Permissible range of values: 00000000..7FF00000 (hexadecimal digits). If necessary, the address is padded with leading zeros.

VM-IDENTIFICATION = <integer 2..99> / <name 1..8>

Identifies the virtual machine by means of a VM index or VM name.

MSCF-COORDINATED =

The (active) VM is stopped by VM2000 (`WAIT(INT)` state, see [page 540](#)), while it is moved in main memory. The duration of this process is not negligible for a large main memory size (greater than 2 Gbyte; more than 10 seconds). For a guest system BS2000/OSD as of V7.0, this operand enables lock mechanisms to be used in the MSCF network to protect the guest system from being regarded as failed by MSCF partners while it is being moved. Detailed information on MSCF is provided in the “HIPLEX MSCF” manual [9].



For guest systems BS2000/OSD V6.0 for which no automatic protection takes place via MSCF, precautions must be taken to ensure that the guest system is not regarded as failed during its move.

This operand is ignored for a VM in the INIT-ONLY or DOWN status.

MSCF-COORDINATED = *IF-POSSIBLE

Protection is requested in the guest system BS2000/OSD as of V7.0. However, the command is executed even if configuration of this protection has not been confirmed after a specified time or has even been rejected.

A VM in the HOLD status or with a guest system other than BS2000/OSD as of V7.0 cannot request protection. The command is executed immediately.

MSCF-COORDINATED = *YES

Protection is requested in the guest system BS2000/OSD as of V7.0. The command is rejected if configuration of this protection has not been confirmed after a specified time or has even been rejected.

The command is not executed for a virtual machine in the HOLD status or with a guest system other than BS2000/OSD as of V7.0.

MSCF-COORDINATED = *NO

Protection is not requested.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
2	0	VMS4103	MSCF protection not (yet) canceled again
	64	VMS4000	VM not yet created
	64	VMS4011	Requested memory size or location not available
	64	VMS4071	Invalid memory size or location
	64	VMS4100	MSCF protection not supported by the guest system (MSCF-COORDINATED=*YES)
	64	VMS4101	MSCF protection rejected by the guest system (MSCF-COORDINATED=*YES)
	64	VMS4102	MSCF protection not confirmed in time by the guest system (MSCF-COORDINATED=*YES)
	128	VMS4049	Memory location of the virtual machine could not be changed as a device is still busy

Example

```
/show-vm-resources information=*memory _____ (1)
```

```
VM-ID      LOC-BEGIN  MEMORY-SIZE
HYPERVISOR 00000000      14 MB
 1 MONITOR  0000000E      128 /   128 MB
 2 SYSTEM2  0000008E      128 /   128 MB
NONE       0000010E      3814 MB
```

```
/move-vm location=*memory-end,vm-id=system2,mscf-coordinated=*if-poss (2)
```

```
/show-vm-resources information=*memory
VM-ID      LOC-BEGIN  MEMORY-SIZE
HYPERVISOR 00000000      14 MB
 1 MONITOR  0000000E      128 /   128 MB
NONE       0000008E      3814 MB
 2 SYSTEM2  00000F74      128 /   128 MB
```

```
/move-vm location=*after-vm(vm-id=monitor),vm-id=system2 _____ (3)
```

```
/show-vm-resources information=*memory
VM-ID      LOC-BEGIN  MEMORY-SIZE
HYPERVISOR 00000000      14 MB
 1 MONITOR  0000000E      128 /   128 MB
 2 SYSTEM2  0000008E      128 /   128 MB
NONE       0000010E      3814 MB
```

- (1) The VM2000 administrator queries the distribution of main memory over the VMs.
- (2) The VM2000 administrator moves VM2 to the end of main memory. A free memory area is created between the monitor VM and VM2. MSCF protection is requested, but the command is executed anyway (default).
- (3) The VM2000 administrator places VM2 behind the monitor VM again.

REDUCE-VM-MEMORY

Reduce main memory for a VM

`/REDUCE-VM-MEMORY` is used to reduce the main memory of a VM during the current session. The reconfiguration of the main memory can be monitored with a monitor job variable.

This command is only permitted for the VM2000 administrator.

SQ On SQ servers the main memory of a VM cannot be reduced while it is being migrated (Live Migration).

Main memory reconfiguration is possible for VMs in the INIT-ONLY or DOWN states. In the case of active guest systems (VM state RUNNING), main memory reconfiguration is only possible after `SYSTEM READY` for the guest system. Main memory reconfiguration is not possible for VMs in the IN HOLD (WAIT) status.

Since dynamic reconfiguration of main memory requires measures to be taken both in the (active) guest system and in VM2000, the process can be time-consuming. The actual main memory reconfiguration takes place asynchronously after the command has been accepted. During this time any further `/EXTEND-VM-MEMORY` or `/REDUCE-VM-MEMORY` command for the same VM will be rejected.

The `/MOVE-VM` (S servers) and `/START-VM` commands can be executed during main memory reconfiguration. In this case, the previous main memory size is used as the main memory size for the relevant VM. The same applies when the guest system is restarted.

REDUCE-VM-MEMORY

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>

,**NEW-MEMORY-SIZE** = ***MINIMUM** / <integer 1..1048576>(**DIMENSION** = ***MEGABYTE** / ***GIGABYTE**)

,**MONJV** = ***NONE** / <filename 1..54 without-gen-vers>

,**JV-PASSWORD** = ***NONE** / <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>

Identifies the virtual machine by means of a VM index or VM name. The monitor VM's main memory can also be reduced.

NEW-MEMORY-SIZE = *MINIMUM / <integer 1..1048576>(DIMENSION = *MEGABYTE / *GIGABYTE)

New size of main memory after reduction for the relevant VM up to a size of 1 Tbyte (terabyte; 1 Tbyte = 1024 Gbytes = 1 048 576 Mbytes).

For a guest system BS2000/OSD V6.0 the value must be less than 32767 Mbytes.

- S On S servers the new main memory size may be below the `MIN-MEMORY-SIZE` of the VM concerned if the VM is in the `INIT-ONLY` or `DOWN` status. In this case the value for the VM's `MIN-MEMORY-SIZE` is assigned the value of `NEW-MEMORY-SIZE`.
- SQ On SQ servers the new main memory size may not be below the `MIN-MEMORY-SIZE` of the VM concerned. The value must be a multiple of 2 Mbytes.

If `*MINIMUM` is specified, the main memory size of the VM is reduced to the smallest possible main memory size (`MIN-MEMORY-SIZE`).

MONJV =

Defines a monitor job variable for monitoring the reconfiguration of the main memory (see [page 272](#)).

MONJV = *NONE

No monitor job variable is defined.

MONJV = <filename 1..54 without-gen-vers>

The specified `MONJV` is supplied with data by VM2000; the sender of the command must therefore have write permission for the specified `MONJV`. If no catalog ID or user ID is specified in the file name, VM2000 completes the file name, in accordance with the runtime environment of the sender of the command. If a file name which does not yet have a job variable is specified, VM2000 creates the `MONJV` as shareable and without a password.

JV-PASSWORD = *NONE / <c-string 1..4> / <x-string 1..8> / <integer ...>

Password of the `MONJV` for checking access authorization. The input field is automatically blanked in the guided dialog. The value input is not logged.



This operand is only evaluated if a previously created monitor job variable was specified with `MONJV=`.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
1	0	CMD0001	No action required
	64	VMS2070	VM in IN HOLD (WAIT) state
	64	VMS4000	VM not yet created
	64	VMS4011	Requested memory size or location not available
	64	VMS4071	Invalid memory size or location
	64	VMS4091	A guest system for which main memory reconfiguration is not possible is running on the VM
	64	VMS4093	Invalid memory size
	64	VMS4340	Opening of MONJV rejected by subsystem JV
	64	VMS4341	JV already assigned to another VM
	128	VMS3042	VM being migrated. Command rejected
	128	VMS4092	Memory reconfiguration is still active

Example

```
/reduce-vm-memory vm-id=2,new-memory-size=*minimum _____ (1)
```

```
/show-vm-resources information=*memory _____ (2)
```

```
VM-ID          LOC-BEGIN    MEMORY-SIZE
HYPERVISOR     00000000      14 MB
 1 MONITOR     0000000E      128 /      128 MB
 2 SYSTEM2     0000008E      256 /      128 /(      128) MB
NONE           0000018E      3686 MB
```

```
/show-vm-resources information=*memory
```

```
VMS4094 MEMORY-RECONFIGURATION FOR VIRTUAL MACHINE (2,SYSTEM2) HAS BEEN
        SUCCESSFULLY PROCESSED. CURRENT MEMORY SIZES: 128 / 128 _____ (3)
```

```
VM-ID          LOC-BEGIN    MEMORY-SIZE
HYPERVISOR     00000000      14 MB
 1 MONITOR     0000000E      128 /      128 MB
 2 SYSTEM2     0000008E      128 /      128 MB _____ (4)
NONE           0000010E      3814 MB
```

- (1) The main memory for the SYSTEM2 VM is to be reduced to its minimum value.
- (2) The VM2000 administrator fetches information on the main memory distribution in VM2000. VM2 has a main memory size of 256 Mbytes and a minimum memory size of 128 Mbytes. The planned new memory size of 128 Mbytes is shown as well while the request is being processed.
- (3) Completion of the asynchronous reconfiguration request is reported to the VM2000 administrator as an asynchronous (guaranteed) message in the user event stream or displayed at the next input (main memory reconfiguration is time-consuming).
- (4) The main memory size for VM2 is now 128 Mbytes (again). This increases the remaining free main memory.

REMOVE-VM-DEVICES

Deallocate devices from VM

`/REMOVE-VM-DEVICES` is used to remove devices explicitly from a VM that were previously assigned explicitly or implicitly. Individual devices are addressed using their mnemonic device names. The devices of a pubset are determined by means of the catalog ID of the pubset or the mnemonic device name of the pubres device.

The home pubset and other imported pubsets, paging disks and the final console cannot be removed from the monitor VM.



The devices removed from a VM should be disconnected beforehand in the corresponding guest system with `/DETACH-DEVICE`.



Alias devices (PAV) cannot be explicitly removed.

REMOVE-VM-DEVICES

```

UNITS = *ALL(...) / *PUBSET-DEVICES(...) /
          list-poss(28): *DEVICE-RANGE(...) / <alphanum-name 1..4 with-wild(20)>

  *ALL(...)
    |   CHECK-VM-STATE = *YES / *NO
  *PUBSET-DEVICES(...)
    |   PUBSET = *BY-PUBRES-DEVICE(...) / <cat-id 1..4>
    |   *BY-PUBRES-DEVICE(...)
    |     |   UNIT = <alphanum-name 2..2> / <x-text 4..4>
  *DEVICE-RANGE(...)
    |   FROM = <alphanum-name 2..2> / <x-text 4..4>
    |   ,TO = <alphanum-name 2..2> / <x-text 4..4>
, VM-IDENTIFICATION = *CURRENT / <integer 1..99> / <name 1..8>
, FORCE = *NO(...) / *YES
  *NO(...)
    |   WAIT-FOR-UNASSIGN = *NO / *YES

```

UNITS =

Defines the devices to be removed from the VM.

UNITS = *ALL(...)

All the devices assigned to this VM are removed. Alias devices are also removed. If a device cannot be removed when *ALL is specified, the command is not executed for this device.

CHECK-VM-STATE =

Determines whether the command is executed, depending on the current state of the VM.

CHECK-VM-STATE = *YES

The command is only executed if the VM is in the DOWN or INIT-ONLY state.

CHECK-VM-STATE = *NO

The command is executed regardless of the VM state.

UNITS = *PUBSET-DEVICES(...)

All devices of a pubset are to be removed from the VM.



See also [“Notes on pubset specification in the VM2000 commands”](#) on page 313.

PUBSET = *BY-PUBRES-DEVICE(...)

Determines the devices of the pubset by means of the pubres device.

UNIT = <alphanum-name 2..2> / <x-text 4..4>

Mnemonic device name of the pubres device.

PUBSET = <cat-id 1..4>

Catalog ID of the pubset. The pubset must be entered together with its pubres device in the monitor system's MRS catalog.

UNITS = *DEVICE-RANGE(...)

All the devices in this name range are to be removed from the VM.



Device names <x-text 4..4> (in *DEVICE-RANGE) are selected from FROM to TO in hexadecimal order. Otherwise, device names <alphanum-name 2..2> are selected in alphabetical order.

If UNITS is specified with *DEVICE-RANGE or with wildcards, alias devices and devices that are not assigned to the VM are ignored.

FROM = <alphanum-name 2..2> / <x-text 4..4>

Defines the start of the name range.

TO = <alphanum-name 2..2> / <x-text 4..4>

Defines the end of the name range. The mnemonic device name must be in the same value range (2- or 4-digit mn) as the FROM device name and must be greater than or equal to the FROM device name in ascending order in the value range.

UNITS = <alphanum-name 1..4 with-wild(20)>

The device is removed from the VM if the VM is in the DOWN or INIT-ONLY state or if the device is detached in the BS2000 guest system except if

VM-IDENTIFICATION =

Identification of the virtual machine.

VM-IDENTIFICATION = *CURRENT

The VM identification assigned previously by means of /BEGIN-VM-DIALOG is used.

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>

Identifies the virtual machine by means of a VM index or VM name.

FORCE =

Specifies the conditions under which devices are removed.

FORCE = *NO(...)

The device is removed from the VM if the VM is in the DOWN or INIT-ONLY state or if the device is detached in the BS2000 guest system. The device is not removed, if there is an outstanding input/output for the device, or if the VM is in the wait state (after /HOLD-VM) or if the guest system is in the startup phase.

SQ On SQ servers no devices can be removed from a VM with FORCE=*NO while it is being migrated (Live Migration).

WAIT-FOR-UNASSIGN = *NO

If the device is still attached in the guest system, it remains assigned explicitly to the VM.

WAIT-FOR-UNASSIGN = *YES

If the device is still attached in the guest system, it remains assigned explicitly to the VM until the BS2000 command /DETACH-DEVICE implicitly releases this device in the guest system. The explicit assignment of the device to the VM is canceled.

FORCE = *YES

VM2000 tries to remove the device from the VM, even if it is still in use. If this is not possible, message VMS4003 or VMS4031 is output. The operand should only be used in exceptional circumstances.

S On S servers the IO-RESET privilege of the VM must be noted (see [page 145](#)).

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
1	0	CMD0001	Pubset not assigned to VM (VMS4236)
1	0	VMS4035	No corresponding device found
1	0	VMS4180	Specified set of devices empty
2	0	CMD0001	Device remains implicitly assigned to the VM until it is detached in the guest system (VMS4087)
2	0	VMS4226	Not all of the pubset's devices could be removed
2	0	VMS4235	Some devices of the pubset remain implicitly assigned to the VM (VMS4087)
2	0	VMS4237	Only the pubres device could be removed
	64	VMS3002	Invalid operand value (e.g. *DEVICE-RANGE operand)
	64	VMS3040	Not processed because of the state of the VM
	64	VMS4000	VM not yet created
	64	VMS4006	Device is not available
	64	VMS4007	Device is not assigned to the virtual machine
	64	VMS4021	Device is the last console or main console of the monitor system. Reconfiguration not possible
	64	VMS4022	Device is currently being used as a public disk in the monitor system. Reconfiguration not possible
	64	VMS4031	Device is INOP
	64	VMS4085	Device not detached in guest system (FORCE=*NO)
	64	VMS4086	Device state not known in the guest system (FORCE=*NO)
	64	VMS4143	Guest system is in the startup phase (FORCE=*NO)
	64	VMS4144	VM is in the wait state (FORCE=*NO)
	64	VMS4190	Device is an alias device (PAV)
	64	VMS4223	Command not executed (operand *PUBSET-DEVICES)
	64	VMS9005	Command could not be processed or could only be partially processed
	128	VMS3042	VM being migrated. Command rejected
	128	VMS4003	Device is busy

Guaranteed messages

VMS4003, VMS4009, VMS4021, VMS4022, VMS4031, VMS4085, VMS4086, VMS4087

Only if *PUBSET-DEVICES is specified: VMS4223, VMS4225, VMS4226, VMS4235, VMS4236, VMS4237.

Only if UNITS is specified without *DEVICE-RANGE and without wildcards: VMS4006, VMS4007 (but not in the case of UNITS specification with *PUBSET-DEVICES), VMS4190.

Example

```

/remove-vm-devices vm-id=system2,
    units=*pubset-devices(pubset=*by-pubres-device(unit=d2)) _____ (1)
VMS4085 DEVICE 'D2' STILL ATTACHED IN GUEST SYSTEM. DEVICE NOT REMOVED
VMS4223 VM2000 COMMAND /REMOVE-VM-DEVICES FOR PUBSET SPECIFIED BY
    PUBRES DEVICE OR CATID 'D2' REJECTED _____ (2)

/remove-vm-devices vm-id=system2,force=*yes
    units=*pubset-devices(pubset=*by-pubres-device(unit=d2)) _____ (3)
VMS4009 DEVICE 'D2' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4225 PUBSET 'VM02' WITH PUBRES DEVICE 'D2' REMOVED FROM VM (2,SYSTEM2)

/remove-vm-devices units=*all(check-vm-state=*no),vm-id=system2,
    force=*yes _____ (4)
VMS4009 DEVICE 'Z2' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4009 DEVICE 'Z3' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4009 DEVICE 'D3' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2) _____ (5)

```

- (1) The VM2000 administrator wants to remove the device(s) of a pubset with the pubres device D2 from VM2.
- (2) The devices cannot be removed because they are still attached in the guest system.
- (3) The devices can still be removed using the `FORCE` operand.
- (4) All the devices are to be removed now from VM2 irrespective of the system state.
- (5) An (active) operating system can then no longer continue to work (INOP message on the BS2000 console of the guest system).

See also the detailed examples for device and pubset management on [page 294](#) and on [page 324](#).

REMOVE-VM-FROM-VM-GROUP

Remove VM from VM group (S servers)

SQ No VM groups are available on SQ servers. The command is rejected with message VMS3011 and with general return code VMS9004.

/REMOVE-VM-FROM-VM-GROUP is used to remove a VM from a VM group.

This command is only permitted for the VM2000 administrator.

REMOVE-VM-FROM-VM-GROUP
VM-IDENTIFICATION = <integer 1..99> / <name 1..8> ,VM-GROUP-NAME = <name 1..8> ,CPU-QUOTA = <fixed 0.01..99.99>

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>

Identifies the virtual machine by means of a VM index or VM name.

VM-GROUP-NAME = <name 1..8>

Name of the VM group.

CPU-QUOTA = <fixed 0.01..99.99>

New CPU quota of the VM.

Determines the VM's own CPU share in comparison to the VM groups and the other VMs which do not belong to a VM group (see [page 238](#)). The VM's previous member CPU quota of the VM is relinquished.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
2	0	CMD0001	Decimal specifications rounded to 2 decimal places
	64	VMS4000	VM not yet created
	64	VMS4427	VM does not belong to VM group

Guaranteed messages

VMS4429

Example

```
/remove-vm-from-vm-group vm-id=c1sys1,vm-group-name=gcust1,  
                        cpu-quota=30 _____ (1)  
VMS4429 VIRTUAL MACHINE (2,C1SYS1) REMOVED FROM VM GROUP 'GCUST1'
```

- (1) The VM C1SYS1 is removed from the VM group. It (again) becomes an autonomous VM with a CPU quota of 30. The member CPU quota is relinquished.

See also the detailed [“Example of operating VM groups”](#) on page 182.

REMOVE-VM-RESOURCES

Cancel assignment of global storage to VM (S servers)

SQ Global storage is not available on SQ servers. The command is rejected with message VMS4114.

/REMOVE-VM-RESOURCES **s** used to cancel an assignment of global storage to a VM made with /ADD-VM-RESOURCES.

REMOVE-VM-RESOURCES
<pre> GS = *REAL / *VIRTUAL(...) *VIRTUAL(...) PARTITION-ID = VIRTGS01 / VIRTGS02 / ... / VIRTGS15 / VIRTGS16 ,VM-IDENTIFICATION = *CURRENT / <integer 1..99> / <name 1..8> ,FORCE = *NO / *YES </pre>

GS =

Defines the part of global storage for which the assignment is to be canceled.

GS = *REAL

The assignment of the entire global storage is canceled.



The real global storage can only be removed from the monitor VM if no partitions of global storage are assigned to other VMs.

GS = *VIRTUAL(PARTITION-ID = VIRTGS01 / VIRTGS02 / ... / VIRTGS15 / VIRTGS16)

Specifies the partition of global storage for which assignment to a VM (virtual assignment) is to be canceled.

VM-IDENTIFICATION =

Identification of the virtual machine.

VM-IDENTIFICATION = *CURRENT

The VM identification assigned previously by means of /BEGIN-VM-DIALOG is used.

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>

Identifies the virtual machine by means of a VM index or VM name.

FORCE =

Defines whether assignment of global storage which is in use is to be canceled or not.

FORCE = *NO

The assignment of global storage is only canceled if the guest system has disconnected from global storage.

FORCE = *YES

The assignment of global storage is canceled even if the guest system has not disconnected from global storage.



Global storage should be removed initially with `FORCE=*NO`. If this is not possible, usage of global storage in the guest system should be terminated. If this is not possible either, global storage must be removed with `FORCE=*YES`.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	64	VMS4000	VM not yet created
	64	VMS4114	Global storage has not been assigned to virtual machine
	64	VMS4115	Invalid GS operand value specified
	64	VMS4116	A virtual machine with virtual global storage exists
	64	VMS4120	VM already connected to global storage

Example (requirements: see [page 434](#))

```

/show-vm-resources information=*gs _____ (1)
VM-ID      PARTITION-ID ATTR  GS-SIZE      MODE  STATUS
1 VM0001   *REAL      SH           16384 MB    DUAL  ASS
2 VM0002   VIRTGS03                      256 MB    DUAL  ASS
3 VM0003   VIRTGS01   SH           512 MB    DUAL  ASS
4 VM0004   VIRTGS01   SH           512 MB    DUAL  ASS
/remove-vm-resources gs=*virtual(partition-id=virtgs03),vm-id=2 _____ (2)
/remove-vm-resources gs=*virtual(partition-id=virtgs01),vm-id=3
/remove-vm-resources gs=*virtual(partition-id=virtgs01),vm-id=4 _____ (3)
/show-vm-resources information=*gs _____ (4)
INDEX  NAME      PARTITION-ID ATTR  GS-SIZE
VM-ID  PARTITION-ID ATTR  GS-SIZE      MODE  STATUS
1 VM0001   *REAL      SH           16384 MB    DUAL  ASS
2 VM0002   *VIRTUAL                      0 MB    DUAL  REM
3 VM0003   *VIRTUAL      SH           0 MB    DUAL  REM
4 VM0004   *VIRTUAL      SH           0 MB    DUAL  REM

```

- (1) The VM2000 administrator fetches information on the allocation of global storage.
- (2) The assignment of partition `VIRTGS03` to VM2 is to be canceled.
- (3) The assignment of the shared partition `VIRTGS01` to VM3 and VM4 is to be canceled.
- (4) Assignment of the global storage partition has been canceled. The attributes of the assignment remain visible, and a new assignment can be made with other attributes (see [page 280](#)).

RESET-VM

Aborts the guest system on a VM

`/RESET-VM` aborts the guest system on a VM, i.e. the virtual CPUs of the VM are stopped. The VM assumes the INIT-ONLY status.

The attributes and resources of the VM remain intact. ADMIN and VC dialogs continue to run. Implicitly assigned devices (see [page 288](#)) are, however, removed from the VM. If main memory reconfiguration is being carried out, it is terminated.

S On S servers the Channel Subsystem Monitoring Facility function is switched off for the guest system.

SQ On SQ servers the guest system on a VM cannot be aborted while the VM is being migrated (Live Migration).

On SQ servers the “Abort VM” function of the SQ manager also aborts the guest system, see the manual “SQ Series Business Servers, Operation and Administration” [18].

The following tasks are facilitated with `/RESET-VM`:

- suppression of disk accesses by a guest system that has apparently failed
- reduction of `MEMORY-SIZE` with `/REDUCE-VM-MEMORY` (if problems occur at command execution during operation)
- reduction of `MIN-MEMORY-SIZE` with `/REDUCE-VM-MEMORY` (S servers)

This command cannot be used on the monitor VM.

RESET-VM
VM-INDEX = <integer 1..99> , VM-NAME = <name 1..8>

VM-INDEX = <integer 1..99>

Specifies the VM index.

VM-NAME = <name 1..8>

Name of the VM whose guest system is to be aborted.



Both operands must be specified to avoid terminating the wrong guest system inadvertently.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
	64	VMS3036	VM name and VM index do not correspond
	64	VMS4000	VM not yet created
	128	VMS3042	VM being migrated. Command rejected

Example

```

/show-vm-attributes vm-id=system2
      CPU-POOL(P)
VM-ID      VM-GROUP(G)  CPU-Q    MAX-U    PROC    MEM-SIZE  MAX-IO
 2 SYSTEM2 *STDPOOL(P) 30.00  100.00    1      1024 MB (100)

VM-ID      STATE          GUEST    VERSION  ADMIN  PRIV
 2 SYSTEM2  RUNNING          BS2-OSD  V09.0A  NO     AS
/reset-vm vm-index=2,vm-name=system2 _____ (1)
VMS3051 VIRTUAL MACHINE (2,SYSTEM2) RESET
% VMS2051 GUEST SYSTEM ON VM (2,SYSTEM2) DOWN;
      REASON: VM-ADMINISTRATOR COMMAND _____ (2)
/show-vm-attributes vm-id=system2
      CPU-POOL(P)
VM-ID      VM-GROUP(G)  CPU-Q    MAX-U    PROC    MEM-SIZE  MAX-IO
 2 SYSTEM2 *STDPOOL(P) 30.00  100.00    1      1024 MB (100)

VM-ID      STATE          GUEST    VERSION  ADMIN  PRIV
 2 SYSTEM2  INIT-ONLY          UNKNOWN          NO     AS _____ (3)
    
```

- (1) The guest system on VM2 which is still active but defective is aborted.
- (2) Asynchronous message about the guest system status.
- (3) The new state of the VM is INIT-ONLY. The VM retains its attributes and resources (with the exception of implicitly assigned devices).

RESTORE-VM-CONFIGURATION

Restore the VM2000 configuration

`/RESTORE-VM-CONFIGURATION` is used to restore the VM2000 configuration from the file in which it was previously backed up. See [section “Backing up and restoring the VM2000 configuration” on page 363](#).

This command is only permitted for the VM2000 administrator.

RESTORE-VM-CONFIGURATION

FROM-FILE = <filename 1..54 without-gen>

FROM-FILE =

Name of the backup file from which the VM2000 configuration is to be restored. The file must have been created using the `/SAVE-VM-CONFIGURATION` command.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
2	0	VMS4802	VM2000 configuration only partially restored from backup file
	64	VMS4015	VM ID already reserved
	64	VMS4202	Assignment set already created
	64	VMS4610	VM group already created
	64	VMS4602	CPU pool already created
	64	VMS4810	VM2000 version too old
	64	VMS4812	Number of real normal or extra CPUs too low
	64	VMS4842	Error reading the backup file
	64	VMS4844	A particular record is missing in the backup file
	64	VMS4845	A particular record is missing in the backup file for a particular component
	64	VMS4846	A particular component is missing in the backup file
	64	VMS4847	Faulty record in the backup file
	64	VMS4848	Faulty component in the backup file
	128	VMS4831	Memory reconfiguration being processed

Guaranteed messages

VMS4801, VMS4802

Example

```
/restore-vm-configuration from-file=vm2000-conf.s200  
VMS4801 VM2000 CONFIGURATION RESTORED FROM FILE :4IVZ:$TSOS.VM2000-CONF.S200
```

RESUME-VM

Resume VM operation

`/RESUME-VM` is used to cancel a VM wait state explicitly imposed beforehand. After command execution, the VM is in the `RUNNING` state, provided no further wait state had to be taken into consideration.

The following rules apply to canceling the wait state:

- The wait state of a VM suspended by the VM administrator can be canceled by the VM2000 administrator
- The wait state of a VM suspended by the VM2000 administrator cannot be canceled by the VM administrator



Owing to “pending interrupts”, the VM can continue at another address, in another state than specified in the corresponding `/HOLD-VM` command.

Canceling overlapping wait states is described on [page 163](#).

RESUME-VM
VM-IDENTIFICATION = <u>*CURRENT</u> / *ALL / <integer 1..99> / <name 1..8>

VM-IDENTIFICATION =

Identification of the virtual machine.

VM-IDENTIFICATION = *CURRENT

The VM identification assigned previously by means of `/BEGIN-VM-DIALOG` is used.

VM-IDENTIFICATION = *ALL

The global wait state is canceled for all suspended VMs. This operand is reserved for the VM2000 administrator.

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>

Identifies the virtual machine by means of a VM index or VM name.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
1	0	CMD0001	No action required
2	0	VMS2020	VM remains in IN HOLD(WAIT) state
	64	VMS2001	VM not in IN HOLD(WAIT) state
	64	VMS4000	VM not yet created
	64	VMS4612	No attached CPU in the CPU pool
	64	VMS9005	Command could not be processed or could only be partially processed

Guaranteed messages

VMS2001, VMS2016, VMS2020, VMS4083

Example

```

/show-vm-attributes vm-id=system2 _____ (1)
VM-ID          VM-GROUP(G) CPU-Q   MAX-U   PROC   MEM-SIZE  MAX-IO
 2 SYSTEM2     *STDPOOL(P) 30.00  100.00   1     1024 MB (100)

VM-ID          STATE                GUEST  VERSION ADMIN PRIV
 2 SYSTEM2     WAIT(SEL)            BS2-OSD V09.0A NO   AS _____ (2)
/resume-vm vm-id=system2 _____ (3)
VMS2016 VIRTUAL MACHINE (2,SYSTEM2) RESUMED BY THE VM2000 ADMINISTRATOR
/show-vm-attributes vm-id=system2
VM-ID          VM-GROUP(G) CPU-Q   MAX-U   PROC   MEM-SIZE  MAX-IO
 2 SYSTEM2     *STDPOOL(P) 30.00  100.00   1     1024 MB (100)

VM-ID          STATE                GUEST  VERSION ADMIN PRIV
 2 SYSTEM2     RUNNING             BS2-OSD V09.0A NO   AS _____ (4)

```

- (1) The VM2000 administrator requests the status of VM2.
- (2) VM2 is in the wait state WAIT(SEL), triggered by a /HOLD-VM VM-ID=2 from the VM2000 administrator.
- (3) The wait state for VM2 is canceled by the VM2000 administrator.
- (4) VM2 is (again) in the RUNNING state.

SAVE-VM-CONFIGURATION

Save the VM2000 configuration

/SAVE-VM-CONFIGURATION is used to save the current VM2000 configuration to a file. See [section “Backing up and restoring the VM2000 configuration” on page 363](#).

This command is only permitted for the VM2000 administrator.

SQ On SQ servers the VM2000 configuration cannot be backed up while a VM is being migrated (Live Migration).

SAVE-VM-CONFIGURATION
TO-FILE = *STD / <filename 1..54 without-cat-gen-ver>

TO-FILE =

Name of the backup file for the current VM2000 configuration.

TO-FILE = *STD

The file is assigned the name SYS.VM2.CFG.<system_name><timestamp>.

TO-FILE = <filename 1..54 without-cat-gen-ver>

Explicitly defines the name of the backup file.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
2	0	VMS4814	An extra CPU is attached when the VM2000 configuration is restored from the backup file
	64	VMS2070	VM in IN HOLD(WAIT) state
	64	VMS4840	Creation of the backup file rejected by DMS
	64	VMS4841	Error writing to the backup file
	128	VMS3042	VM being migrated. Command rejected
	128	VMS4830	VM has not yet reached SYSTEM READY
	128	VMS4831	Memory reconfiguration being processed
	128	VMS4832	Device configuration during processing

Guaranteed messages

VMS4800

Example

```
/save-vm-configuration to-file=vm2000-conf.s200  
VMS4800 CURRENT VM2000 CONFIGURATION SAVED IN FILE :4IVZ:$TSOS.VM2000-  
CONF.S200
```

SEND-VM-MESSAGE

Pass messages to VM administrators

/SEND-VM-MESSAGE enables messages to be sent to the VM administrators.

SEND-VM-MESSAGE

MESSAGE = <c-string 1..72>

,VM-IDENTIFICATION = *ALL / <integer 1..99> / <name 1..8>

MESSAGE = <c-string 1..72>

Message to be sent to all VM administrators of the specified VM.

VM-IDENTIFICATION =

Identification of the virtual machine.

VM-IDENTIFICATION = *ALL

The message is sent to all VM administrators of all VMs.

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>

Identifies the virtual machine by means of a VM index or VM name.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
	64	VMS1015	No VM administrator connected
	64	VMS4000	VM not yet created

Example (via \$VMCONS)

```

vm01:/send-vm-message message=c'*** shutdown in 5 min ***',vm-id=*all — (1)
VM01:VMS3000 VM2000 COMMAND /SEND-VM-MESSAGE FOR
          VIRTUAL MACHINE ( ,*ALL) PROCESSED SUCCESSFULLY
VM01:NBR0740 COMMAND COMPLETED 'SEND-VM-MESSAGE';
          (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
VM01:VMS2036 MESSAGE FROM VIRTUAL MACHINE (1,VM1):
          '*** SHUTDOWN IN 5 MIN ***' _____ (2)
VM02:VMS2036 MESSAGE FROM VIRTUAL MACHINE (1,VM1):
          '*** SHUTDOWN IN 5 MIN ***' _____ (3)
VM04:VMS2036 MESSAGE FROM VIRTUAL MACHINE (1,VM1):
          '*** SHUTDOWN IN 5 MIN ***' _____ (4)
vm04:/send-vm-message message=c'** shutdown !!! **',vm-id=2 _____ (5)
VM04:VMS3000 VM2000 COMMAND /SEND-VM-MESSAGE FOR
          VIRTUAL MACHINE (2,SYSTEM2) PROCESSED SUCCESSFULLY
VM04:NBR0740 COMMAND COMPLETED 'SEND-VM-MESSAGE';
          (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
VM02:VMS2036 MESSAGE FROM VIRTUAL MACHINE (4,SYSTEM4):
          '** SHUTDOWN !!! **' _____ (6)

```

- (1) The VM2000 administrator sends the message ***** SHUTDOWN IN 5 MIN ***** to all VM administrators.
- (2) This message appears in the ADMIN dialog for the monitor VM.
- (3) This message appears in the ADMIN dialog for VM2.
- (4) This message appears in the ADMIN dialog for VM4.
- (5) The VM4 administrator sends the message **** SHUTDOWN !!! **** to the VM2 administrator. ****** to the VM2 administrator.
- (6) This message appears in the ADMIN dialog for VM2.



When administration is carried out from a user task, message VMS2036 is displayed as an asynchronous message in the user event stream or at the next input.

SHOW-VM-ASSIGNMENT-SET

Output assignment sets

/SHOW-VM-ASSIGNMENT-SET is used to output the attributes of assignment sets.

The command supplies the following information:

- Name of the assignment set
- VMs which devices from the relevant assignment set may implicitly assign themselves
- Number of devices assigned to the assignment set

This command is only permitted for the VM2000 administrator.

SHOW-VM-ASSIGNMENT-SET
SET-NAME = *ALL / *STDSET / <name 1..8>

SET-NAME =

Determines the assignment set about which information is to be output.

SET-NAME = *ALL

Information is to be output about all assignment sets.

SET-NAME = *STDSET / <name 1..8>

Name of the assignment set.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
	64	VMS4200	Assignment set not yet created

Output in S variables (in user task)

Output information	Name of the S variable	T	Contents	Condition
Name of the assignment set	var(*LIST).ASSIGNMENT-SET	S	*STDSET <name 1..8>	
Index of the VM which may implicitly assign itself devices from this assignment set	var(*LIST).VM(*LIST).VM-INDEX	I	<integer 1..99>	
Name of the VM which may implicitly assign itself devices from this assignment set	var(*LIST).VM(*LIST).VM-NAME	S	<name 1..8>	
Number of devices which are assigned to the assignment set	var(*LIST).NUM-OF-DEV	I	<integer 0..32767>	
Are new devices added to this assignment set?	var(*LIST).NEW-DEV	B	TRUE / FALSE	
Number of VMs which may implicitly assign themselves devices from this assignment set	var(*LIST).NUM-OF-VMS	I	<integer 0..99>	

Output format*Example*

ASSIGNMENT-SET	#DEVICES	VM-ID
*STDSET	2+NEW	1 MONITOR 3 SYSTEM3
DUMMY	0	NONE
SYS2IMPL	3	2 SYSTEM2

Meaning of the output columns

ASSIGNMENT-SET

Name of the assignment set.

#DEVICES

Number of devices which are assigned to the assignment set.

+NEW

Suffix which indicates that new devices are assigned to this assignment set.

VM-ID

Identifies a VM which may implicitly assign itself devices from this assignment set by means of the VM index and VM name.

NONE

No VM may implicitly assign itself devices from this assignment set.

Example

See also the detailed examples for device and pubset management on [page 294](#) and on [page 324](#).

SHOW-VM-ATTRIBUTES

Output attributes of a VM

Using `/SHOW-VM-ATTRIBUTES`, the attributes of virtual machines are output.

The command supplies the following information:

- the attributes of the VM, as defined by means of command
- the VM status
- the name and version of the operating system on the guest system
- whether a VM administrator is connected to the VM
- the privileges for the VM
- the VM group to which the VM belongs
- the CPU pool to which the VM is assigned
- the scheduling procedure of the VM
- the settings of the VM for controlling the real CPU
- the state of the virtual CPUs
- the devices assigned to the VM
- the global storage assigned to the VM
- start address and size of the VM's main memory (in Mbytes)
- the monitor job variables assigned to the VM

SHOW-VM-ATTRIBUTES

```

INFORMATION = *STD / *CPU / *UNITS / *GS / *MEMORY / *MONJV / *PRIVILEGES / *ALL
,TYPE = *ALL / *SD / *VC / <family> / <device>
,VM-IDENTIFICATION = *CURRENT / *VM2000 / <integer 1..99> / <name 1..8 with-wild(20)> /
                    *BY-VM-GROUP(...) / *BY-CPU-POOL(...)
    *BY-VM-GROUP(...)
    |   VM-GROUP-NAME = <name 1..8>
    *BY-CPU-POOL(...)
    |   CPU-POOL-NAME = *STDPOOL / <name 1..8>

```

INFORMATION =

This defines the output information.

INFORMATION = *STD

Only VM attributes and statuses.

INFORMATION = *CPU

Only information on the state of the virtual CPUs of the VM.

INFORMATION = *UNITS

Only the devices which are assigned to the VM are output.

INFORMATION = *GS

Only information about global storage assigned to the VM.

INFORMATION = *MEMORY

Only information on the VM's main memory.

INFORMATION = *MONJV

Only information on the monitor job variables which are assigned to the VM.

INFORMATION = *PRIVILEGES

Only information on the privileges of the VM.

INFORMATION = *ALL

All the information contained in the outputs *STD, *CPU, *UNITS, *GS, *MEMORY, *MONJV and *PRIVILEGES.

TYPE =

Selects the devices assigned to the VM that are to be output. This operand is evaluated only if the operand `INFORMATION=*ALL/UNITS` is specified at the same time.

TYPE = *ALL

Output of all devices assigned to the VM.

TYPE = *SD

Output of all devices of the VM which are declared as shareable.

TYPE = *VC

Output of all devices of the VM which are assigned as virtual console.

TYPE = <family>

Output of the devices of the specified device family which are assigned to the VM.

<family>	<family> name
*CONSOLE	Console
*PRINTER	Printer
*FAM50	Special devices
*TD	TRANSDATA device
*EXOTIC	Physically supported device (ADAM)
*UR	Unit record device
*DISK	Disk device
*UNMTAPE	Unimodal tape device
*MBK	Magnetic tape cartridge device
*BIMTAPE	Bimodal tape device
*TAPE	Tape device

TYPE = <device>

Output of all devices of the specified device type which are assigned to the VM. The possible device types are contained in the device type table (Device type column) in the "System Installation" manual [1]. The possible device and volume types are displayed in the dialog with `TYPE=?`. You can find out the device types present on the current system with, for example, `/SHOW-DEVICE-CONFIGURATION`.

VM-IDENTIFICATION =

Identification of the virtual machine.

VM-IDENTIFICATION = *CURRENT

The VM identification assigned previously by means of /BEGIN-VM-DIALOG is used.

VM-IDENTIFICATION = *VM2000

The information is output for all configured VMs. This operand is reserved for the VM2000 administrator.

VM-IDENTIFICATION = <integer 1..99> / <name 1..8 with-wild(20)>

Identifies the virtual machine by means of a VM index or VM name. Only the VM2000 administrator is permitted to enter wildcards in the VM name or specify an arbitrary VM.

VM-IDENTIFICATION = *BY-VM-GROUP(...)

Information is output for all VMs which belong to the VM group (S servers). This operand is reserved for the VM2000 administrator.

VM-GROUP-NAME = <name 1..8>

Name of the VM group.

VM-IDENTIFICATION = *BY-CPU-POOL(...)

Information is output for all VMs which belong to the CPU pool. This operand is reserved for the VM2000 administrator.

CPU-POOL-NAME = *STDPOOL / <name 1..8>

Name of the CPU pool.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
1	0	VMS4035	Specified set of devices empty (INFORMATION=*ALL / *UNITS)
2	0	VMS4001	No VM with appropriate name found
2	0	VMS4403	No VM belongs to VM group
2	0	VMS4613	No VM has been assigned to CPU pool
	64	VMS4000	VM not yet created
	64	VMS4400	VM group not yet created
	64	VMS4600	CPU pool not yet created

Output in S variables (in user task)

The `INFORMATION` operand of the command defines which S variables are supplied with values. The following can be specified for `INFORMATION`:

Notation of command	Abbreviated form used in table
<code>INFORMATION = *STD</code>	1
<code>INFORMATION = *CPU</code>	2
<code>INFORMATION = *UNITS</code>	3
<code>INFORMATION = *MEMORY</code>	4
<code>INFORMATION = *GS</code>	5
<code>INFORMATION = *MONJV</code>	6
<code>INFORMATION = *PRIVILEGES</code>	7
<code>INFORMATION = *ALL</code>	8

Output information	Name of the S variable	T	Contents	Condition
Index of VM	<code>var(*LIST).VM-INDEX</code>	I	<integer 1..99>	1 - 8
Name of VM	<code>var(*LIST).VM-NAME</code>	S	<name 1..8>	1 - 8
multiprocessor level of the VM	<code>var(*LIST).NUM-OF-CPUS</code>	I	<integer 1..32>	1,2,8
Number of virtual spare CPUs of the VM	<code>var(*LIST).NUM-OF-VM-SPARE-CPUS</code>	I	<integer 0..15>	1,2,8
Name of the CPU pool	<code>var(*LIST).CPU-POOL</code>	S	*STDPOOL <name 1..8>	1,2,8
CPU quota of the VM (integer)	<code>var(*LIST).CPU-QUOTA</code>	I	<integer 1..99>	1,2,8
CPU quota of the VM (exact)	<code>var(*LIST).FINE-CPU-QUOTA</code>	S	<fixed 0.01..99.99>	1,2,8
Effective CPU share of the VM	<code>var(*LIST).EFFECTIVE-CPU-QUOTA</code>	S	<fixed 0.01..100.00>	1,2,8
Maximum CPU utilization of the VM (integer)	<code>var(*LIST).MAX-CPU-UTILIZATION</code>	I	<integer 1..100>	1,2,8
Maximum CPU utilization of the VM (exact)	<code>var(*LIST).FINE-MAX-UTIL</code>	S	<fixed 0.01..100.00>	1,2,8
Maximum CPU utilization of the VM (effective)	<code>var(*LIST).EFFECTIVE-MAX-UTIL</code>	S	<fixed 0.01..100.00>	1,2,8
Name of the VM group	<code>var(*LIST).VM-GROUP</code>	S	*NONE <name 1..8>	1,2,8
Size of VM main memory in Mbytes	<code>var(*LIST).MEM-SIZE</code>	I	<integer 1..1048576>	1,4,8
Minimum size of VM main memory in Mbytes	<code>var(*LIST).MIN-MEM-SIZE</code>	S	*UNDEF <integer 1..1048576>	1,4,8
New main memory size for VM for main memory reconfiguration in Mbytes	<code>var(*LIST).PLANNED-MEM-SIZE</code>	S	*UNDEF <integer 1..1048576>	1,4,8
Is the IO-RESET privilege assigned?	<code>var(*LIST).IO-PRIVIL</code>	B	TRUE / FALSE	1,7,8

Output information	Name of the S variable	T	Contents	Condition
Is the privilege IO-RESET assigned?	var(*LIST).IO-PRIO-PRIVIL	B	TRUE / FALSE	1,7,8
Is the ASSIGN-BY-GUEST privilege assigned for at least one assignment set?	var(*LIST).ASS-BY-GUEST-PRIVIL	B	TRUE / FALSE	1,7,8
Is the AUTO-SNAP-ASSIGNMENT privilege assigned?	var(*LIST).AUTO-SNAP-PRIVIL	B	TRUE / FALSE	1,7,8
Maximum I/O utilization of the VM	var(*LIST).MAX-IO-UTILIZATION	I	<integer 1..100>	1.8
Is the I/O utilization monitored by IOLVM?	var(*LIST).MAX-IO-UTIL-ACTIVE	B	TRUE / FALSE	1.8
Status of VM	var(*LIST).STA	S	*INIT-ONLY / *RUN *WAIT / *DOWN / *MIGRATING	1.8
Is the VM in the wait state following a global /HOLD-VM from the VM2000 administrator?	var(*LIST).WAIT-GLOBAL	B	TRUE / FALSE	1.8
Is the VM in the wait state following a selective /HOLD-VM from the VM2000 administrator?	var(*LIST).WAIT-SEL	B	TRUE / FALSE	1.8
Is the VM in the wait state following a /HOLD-VM from the VM2000 administrator?	var(*LIST).WAIT-BY-VM-ADM	B	TRUE / FALSE	1.8
Is the VM in the VM2000-internal wait state?	var(*LIST).WAIT-INTERNAL	B	TRUE / FALSE	1.8
Is a VM administrator connected?	var(*LIST).VM-ADM	B	TRUE / FALSE	1.8
Guest operating system	var(*LIST).GUEST-TYPE	S	BS2-OSD *UNKNOWN	1.8
Version of the guest operating systems	var(*LIST).GUEST-VERSION	S	<c-string 1..8> *UNKNOWN	1.8
Scheduling procedures	var(*LIST).SCHEDULE	S	*NONE *BY-TIMESLICE *DEDICATED-CPUS	2.8
CPU quota (exact) of the VM group when the VM is assigned to a VM group	var(*LIST).VM-GROUP-CPU-QUOTA	S	*NONE <fixed 0.01..99.99>	2.8
Virtual CPUs of the VM	var(*LIST).VIRT-CPU(*LIST).NUM	S	<x-text 2..2>	2.8
Status of virtual CPUs of VM	var(*LIST).VIRT-CPU(*LIST).STA	S	*RUN / *IDLE / *INIT *WAIT / *HALT / *STOP *BLOCK / *SLEEP *OFFLINE	2.8

Output information	Name of the S variable	T	Contents	Condition
Setting for VM-ACTIVE-IDLE	var(*LIST).VM-ACTIVE-IDLE	S	*NO *AT-DEDICATED-CPUS	2.8
Devices of the VM	var(*LIST).UNIT(*LIST).MN	S	<alphanum-name 2..2> <x-text 4..4>	3.8
Who assigned the device to the VM?	var(*LIST).UNIT(*LIST).ASS-BY	S	*VM-ADMIN *GUEST	3.8
Usage mode	var(*LIST).UNIT(*LIST).USAGE	S	*EXCL / *SHARED	3.8
Is the device a virtual console?	var(*LIST).UNIT(*LIST).VC	B	TRUE / FALSE	3.8
Is the device in KEEP mode?	var(*LIST).UNIT(*LIST).KEEP	B	TRUE / FALSE	3.8
Start address of VM main memory in Mbytes	var(*LIST).LOCATION	S	<x-text 1..8> *SPREAD	4.8
Maximum size of the VM's main memory in Mbytes	var(*LIST).MAX-MEM-SIZE	S	<integer 1..1048576> *UNDEF	4.8
Size of the main memory for BS2000/OSD in Mbytes	var(*LIST).BS2-MEM-SIZE	I	<integer 1..1048576>	4.8
Partition of global storage assigned to the VM	var(*LIST).PART-ID	S	<name 1..8> / *REAL *VIRTUAL / *NONE	5.8
Usage mode of the global storage of the VM	var(*LIST).GS-ATTR	S	*SHARE / *EXCL *UNDEF	5.8
Size of VM global storage in Mbytes	var(*LIST).GS-SIZE	I	<integer 0..134 217 728>	5.8
Mode of the global storage of the VM	var(*LIST).GS-MODE	S	*MONO / *DUAL *NONE	5.8
State of the global storage of the VM	var(*LIST).GS-STA	S	*ASSIGN / *REMOVE *NONE	5.8
Monitor job variable for monitoring the VM	var(*LIST).VM-MONJV	S	<name 1..54> / *NONE	6.8
Monitor job variable for monitoring the reconfiguration of the main memory for the VM	var(*LIST).RECONF-MONJV	S	<name 1..54> / *NONE	6.8
Command/function scope for VM2000 administrator (monitor VM) and VM administrator	var(*LIST).ADM-PRIVIL	S	*VM2000-MIN / *VM2000-STD / *VM-STD / *VM-MAX	7.8
Specification of the assignment sets for the VM ASSIGN-BY-GUEST=*YES(...)	var(*LIST).ASSIGNMENT-SET-SPEC	S	*NONE / *ALL / *EXCEPT / *LIST	7.8
List of the assignment sets for the VM (only *EXCEPT / *LIST)	var(*LIST).ASSIGNMENT-SET(*LIST)	S	*STDSET / <name 1..8>	7.8

Output format for INFORMATION = *STD*Example*

```

          CPU-POOL(P)
VM-ID    VM-GROUP(G) CPU-Q  MAX-U  PROC  MEM-SIZE  MAX-IO
  2 SYSTEM2 *STDPOOL(P) 30.00 100.00 2(+1)   128 MB (100)

VM-ID    STATE          GUEST  VERSION ADMIN PRIV
  2 SYSTEM2  RUNNING          BS2-OSD V09.0A  NO   AG,AS

```

Meaning of the output columns

VM-ID

Identifies the VM by means of VM index and VM name.

CPU-POOL(P)

VM-GROUP(G)

Shows whether the VM is assigned to a VM group. If the VM is not assigned to a VM group, the CPU pool to which the VM is assigned is shown. Whether the VM is assigned to a VM group is indicated by a suffix:

<vm-group-name>(G) The VM is assigned to a VM group; the VM's member CPU quota is displayed under CPU-Q (S servers).

<cpu-pool-name>(P) The VM is not assigned to a VM group; the CPU quota of the (individual) VM is displayed under CPU-Q.

CPU-Q

CPU quota of the VM as set by means of command. For VMs which belong to a VM group this is the member CPU quota (MEMBER-CPU-QUOTA operand), otherwise the CPU quota of the VM which does not belong to a VM group (CPU-QUOTA operand).

MAX-U

Maximum CPU utilization of the VM as set by means of command (MAX-CPU-UTILIZATION operand).

PROC

Virtual CPUs of the VM in the form *m* or *m(+n)*.

m Number of virtual normal CPUs of the VM (multiprocessor level of the VM)

n Number of virtual spare CPUs of the VM

MEM-SIZE

Current main memory size of the VM in Mbytes.

MAX-IO

Maximum I/O utilization of the VM as set by means of a command (MAX-IO-UTILIZATION operand). If the value is specified in parentheses, this means the VM is currently not being monitored by the IORM function IOLVM, see [page 360](#).

STATE

State of the VM. States may be combined by means of a plus sign (e.g. WAIT(GLB+VMA)).

INIT-ONLY	The VM is initialized (after /CREATE-VM or /RESET-VM), but no guest system has yet been started
RUNNING	The guest system on the VM has been started and is running
WAIT(...)	The VM is in the wait state; the guest system is at rest (cause)
WAIT(GLB)	Global HOLD (/HOLD-VM *ALL) of the VM2000 administrator
WAIT(INT)	Internal wait state of the VM, e.g. during /MOVE-VM
WAIT(SEL)	Selective HOLD (/HOLD-VM VM-ID=2..n) of the VM2000 administrator
WAIT(VMA)	/HOLD-VM of the VM administrator for his VM
DOWN	The guest system on the VM is terminated (after shutdown or SETS in the guest system)
MIGRATING	A live migration (SQ servers) is running for the VM.

GUEST

Operating system in the guest system (e.g. BS2-OSD). UNKNOWN: The operating system is not (yet) known (e.g. in the startup phase).

VERSION

Version of the operating system in the guest system in the form Vnn.na. Blanks are output here if the operating system is not (yet) known (UNKNOWN in GUEST column).

ADMIN

Indicates if a VM administrator is connected to the virtual machine (YES/NO).

PRIV

Privileges of the VM (AG=ASSIGN-BY-GUEST / IO-R=IO-RESET /
AS=AUTO-SNAP-ASSIGNMENT / IO-P=IO-PRIORITY).

Output format for INFORMATION = *CPU*Example*

```
VM-ID          CPU-POOL  GR-CPU-Q  CPU-Q/ EFF-Q  MAX-U/ EFF-U  PROC
 1 MONITOR     *STDPool          30.00/ 50.00 100.00/ 50.00 2(+1)
```

```
VM-ID          SCHED ACT-I  VIRT CPUS
 1 MONITOR     DC      NO      00(RUN),01(RUN),02(OFF)
```

Meaning of the output columns

VM-ID / PROC

Corresponds to INFORMATION=*STD.

CPU-POOL

Shows the CPU pool to which the VM is assigned.

GR-CPU-Q

Shows whether the VM is assigned to a VM group.

If the VM is assigned to a VM group, the CPU quota of the VM group is displayed here, and the member CPU quota of the VM under CPU-Q.

If the VM is not assigned to a VM group, blanks are displayed here, and the CPU quota of the (individual) VM under CPU-Q.

CPU-Q/EFF-Q

CPU quota of the VM in the format <cpu-quota> / <effective-cpu-quota>, **where:**

<cpu-quota>

CPU quota of the VM as set by means of command. For VMs which belong to a VM group this is the member CPU quota (MEMBER-CPU-QUOTA operand), otherwise the CPU quota of the VM which does not belong to a VM group (CPU-QUOTA operand).

<effective-cpu-quota>

Effective CPU share of the VM, calculated by VM2000. Values < 0.01 are rounded up to 0.01.

MAX-U/EFF-U

Maximum CPU utilization of the VM in the format <max-cpu-util> / <effective-max-cpu-util>, **where:**

<max-cpu-util>

Maximum CPU utilization set by means of command (MAX-CPU-UTILIZATION operand) for the VM.

<effective-max-cpu-util>

Effective maximum CPU utilization of the VM, calculated by VM2000.

SCHED

Scheduling procedure:

- There are no virtual CPUs in the scheduling procedure
- DC Fixed CPU assignment (dedicated CPUs)
- TS Time slicing

ACT-I

Setting for controlling the real CPU in the case of fixed CPU assignment:

- NO The real CPU is removed from the VM if the VM's virtual CPU which runs on this is inactive (interruptible wait state, "Idle").
- AT-DC The VM retains control over the real CPU even if the VM's virtual CPU which runs on this is inactive (interruptible wait state, "Idle", S servers).

VIRT CPUS

Virtual normal and spare CPUs of the VM in the format <cpu-id>(<vstate>),

<cpu-id> CPU identifier.

<vstate> State of the virtual CPU, see [page 216](#)

(RUN / IDLE / INIT / WAIT / STOP / HALT / BLOCK / SLEEP / OFF).

Output format for INFORMATION = *UNITS

Example

```
VM-ID          UNITS
 2 SYSTEM2    ASSIGNED BY VM-ADMIN
              EXCL:  CH,CI,505F,5060
              ASSIGNED BY GUEST
              EXCL:  P2,P3,R2,R3
              SHARED: 505A,505B,505C
```

Meaning of the output columns

VM-ID

Identifies the VM by means of VM index and VM name.

UNITS

Mnemonic device names in blocks:

first the explicitly assigned devices (ASSIGNED BY VM-ADMIN), then the implicitly assigned devices (ASSIGNED BY GUEST). The output is grouped in accordance with the usage mode (EXCL/SHARED/KEEP). Empty blocks or groups are omitted.

The device list is arranged in ascending order of the value range (two- or four-digit). The two-digit device names are output first, followed by the four-digit device names.

- Ⓢ The suffix (VC) identifies a virtual console (see [page 117](#)).

Output format for INFORMATION = *GS*Example*

VM-ID	PARTITION-ID	ATTR	GS-SIZE	MODE	STATUS
2 VM0002	VIRTGS03		256 MB	DUAL	ASS

Meaning of the output columns

VM-ID

Identifies the VM by means of VM index and VM name.

PARTITION-ID

Describes the assigned partition.

*REAL

The entire global storage is assigned in the real sense.

*VIRTUAL

No global storage is currently assigned to the VM. But a global storage partition was assigned to the VM virtually beforehand.

NO GS AVAILABLE

No global storage is available at the server.

NO GS ASSIGNED

Global storage is available at the server, but no global storage has been assigned to the VM.

ATTR

Usage mode of the global storage.

SH The global storage can be used by several VMs simultaneously (VMSHARED=*YES)

┘ The global storage is used by the VM exclusively (VM-SHARED=*NO)

GS-SIZE

Size of the real global storage assigned or of the partition in Mbytes.

MODE

Mode of the partition (MONO/DUAL, see CREATE-GS-PARTITION).

STATUS

Specifies whether the global storage described is still assigned to the VM (ASSIGNED) or has been removed (REMOVED).

Output format for INFORMATION = *MEMORY (S servers)*Example*

```
VM-ID          LOC-BEGIN    MEMORY-SIZE
 2 SYSTEM2    0000008E      128 /      128 MB
```

Meaning of the output columns

VM-ID

Identifies the VM by means of VM index and VM name.

LOC-BEGIN

Start address of the main memory area in Mbytes.

MEMORY-SIZE

Main memory of the VM in the form <memory-size> / <min-memory-size> / (<planned-memory-size>) in Mbytes. (<planned-memory-size>) is only output when main memory reconfiguration is in progress.

Output format for INFORMATION = *MEMORY (SQ servers)*Example*

```
VM-ID          BS2-MEM-S  MEM-SIZE  MIN-MEM-S  MAX-MEM-S  DEST-MEM-S
 2 SYSTEM2      970      1024      1024      2048
```

Meaning of the output columns

VM-ID

Identifies the VM by means of VM index and VM name.

BS2-MEM-S

Size of the main memory (without the VM's firmware component) which is available for BS2000/OSD in Mbytes.

MEM-SIZE

Size of the VM's main memory in Mbytes.

MIN-MEM-S

Minimum size of the VM's main memory in Mbytes (for main memory reconfiguration).

MAX-MEM-S

Maximum size of the VM's main memory in Mbytes (for main memory reconfiguration).

DEST-MEM-S

Planned size of the VM's main memory in Mbytes when main memory reconfiguration is in progress.

Output format for INFORMATION = *MONJV*Example*

```
VM-ID          VM-MONJV
 2 SYSTEM2     :4IVZ:$TSOS.JV-SYS2

VM-ID          RECONF-MONJV
 2 SYSTEM2     :4IVZ:$TSOS.JV-S2REC
```

Meaning of the output columns

VM-ID

Identifies the VM by means of VM index and VM name.

VM-MONJV

File name of the MONJV for the VM.

NO VM-MONJV ASSIGNED No MONJV is defined for the MONJV.

RECONF-MONJV

File name of the MONJV for the main memory reconfiguration. If no reconfiguration of the main memory is in progress or no MONJV has been defined for this, the second output block is not output.

Output format for INFORMATION = *PRIVILEGES*Example*

```

VM-ID      PRIV    DETAILS
 2 SYSTEM2 AG      SYS2IMPL
           AS

```

Meaning of the output columns

VM-ID

Identifies the VM by means of VM index and VM name.

PRIV

Privileges of the VM.

NO PRIVILEGE No privileges.

ADM Command/function scope for VM2000 administrator (monitor VM) or VM administrator, see DETAILS.

AG The VM has the ASSIGN-BY-GUEST privilege. The list of the assignment sets assigned is output in DETAILS.

AS AUTO-SNAP-ASSIGNMENT=*YES (default).

IO-P IO-PRIORITY=*YES.

IO-R IO-RESET=*YES.

DETAILS

Supplementary information on the privileges:**For PRIV=ADM**

*VM2000-MIN ADMIN-PRIVILEGES=*MIN for the monitor VM

*VM-MAX ADMIN-PRIVILEGES=*MAX for a VM other than the monitor VM

┆ Standard command/function scope

For PRIV=AG

*ALL ASSIGN-BY-GUEST=*YES(SET-NAME=*ALL)

*EXCEPT: set1, set2, ...

ASSIGN-BY-GUEST=*YES(SET-NAME=*EXCEPT(...))

set1, set2, ...

ASSIGN-BY-GUEST=*YES(SET-NAME=<list>)

Example (S servers)

```

/show-vm-attributes information=*all _____ (1)
      CPU-POOL(P)
VM-ID      VM-GROUP(G) CPU-Q    MAX-U    PROC    MEM-SIZE  MAX-IO
 2 SYSTEM2 *STDPOOL(P) 30.00   100.00   1(+1)   128 MB (100) _____ (2)

VM-ID      STATE          GUEST   VERSION  ADMIN  PRIV
 2 SYSTEM2 RUNNING        BS2-OSD V09.0A  NO    AG,AS _____ (3)

VM-ID      CPU-POOL  GR-CPU-Q  CPU-Q/  EFF-Q  MAX-U/  EFF-U  PROC
 2 SYSTEM2 *STDPOOL          30.00/  33.33  100.00/ 33.33  1(+1)

VM-ID      SCHED ACT-I  VIRT CPUS
 2 SYSTEM2 DC     NO    00(RUN),01(SLEEP) _____ (4)

VM-ID      VM-MONJV
 2 SYSTEM2 NO VM-MONJV ASSIGNED _____ (5)

VM-ID      UNITS
 2 SYSTEM2 ASSIGNED BY VM-ADMIN _____ (6)
      EXCL: CH,CI,E059,E05A,E05B
      ASSIGNED BY GUEST
      EXCL: V2,V3
      SHARED: E057,E058,E062

VM-ID      PARTITION-ID ATTR    GS-SIZE    MODE  STATUS
 2 SYSTEM2 NO GS ASSIGNED _____ (7)

VM-ID      LOC-BEGIN  MEMORY-SIZE
 2 SYSTEM2 0000008E      128 / 128 MB _____ (8)

VM-ID      PRIV  DETAILS
 2 SYSTEM2 AG    SYS2IMPL _____ (9)
      AS

```

- (1) The VM administrator queries the attributes and status of his/her VM.
- (2) VM2 is a monoprocessor VM (one virtual normal CPU) with a virtual spare CPU. The VM2 (VM name SYSTEM2) main memory size is 128 Mbytes. The VM belongs to the standard CPU pool.
- (3) The VM is in the RUNNING state. The guest system is active. No VM administrator is connected. The VM has the ASSIGN-BY-GUEST and AUTO-SNAP-ASSIGNMENT (default) privileges.

- (4) The VM is currently running with fixed CPU assignment (SCHED DC). The real CPU is switched from the VM if the VM's virtual CPU 00 which runs on it is inactive (ACT-I: NO). The virtual normal CPU 00 of the VM is running. The virtual spare CPU 01 is in a wait state in order to be able to take on the job of the virtual normal CPU in the event of the latter's failure.
- (5) No MONJV is assigned to the VM.
- (6) List of mnemonic device names assigned to the VM.
- (7) No global storage is assigned to the VM.
- (8) The start address of the VM domain is X'8E' Mbytes, its size 128 Mbytes.
- (9) The privileges of the VM.

SHOW-VM-CPU-POOL

Output attributes of a CPU pool

/SHOW-VM-CPU-POOL is used to output the attributes of CPU pools.

This command is only permitted for the VM2000 administrator.

The command supplies the following information:

- Name of the CPU pool as defined with /CREATE-VM-CPU-POOL
- The scheduling procedure
- The CPUs in the CPU pool and their status
- The VMs assigned to the CPU pool and their multiprocessor level
- On SQ servers: the static CPU pools *POOL0 and *FOREIGN, see [page 34](#)

SHOW-VM-CPU-POOL

```

CPU-POOL-NAME = *ALL / *STDPOOL / <name 1..8> / *BY-CPU-IDENTIFICATION(...) /
                *BY-VM-IDENTIFICATION(...) / *BY-VM-GROUP(...)

*BY-CPU-IDENTIFICATION(...)
  | CPU-IDENTIFICATION = <x-text 1..2>

*BY-VM-IDENTIFICATION(...)
  | VM-IDENTIFICATION = <integer 1..99> / <name 1..8>

*BY-VM-GROUP(...)
  | VM-GROUP-NAME = <name 1..8>

,INFORMATION = *CPU / *VM / *SUMMARY / *ALL

```

CPU-POOL-NAME =

Determines the CPU pool about which information is to be output.

CPU-POOL-NAME = *ALL

Information about all CPU pools is to be output.

CPU-POOL-NAME = *STDPOOL / <name 1..8>

Name of the CPU pool.

CPU-POOL-NAME = *BY-CPU-IDENTIFICATION(...)

Determines the CPU pool by specifying a CPU contained in the CPU pool.

CPU-IDENTIFICATION = <x-text 1..2>

CPU identifier.

CPU-POOL-NAME = *BY-VM-IDENTIFICATION(...)

Determines the CPU pool by specifying a VM which is assigned to the CPU pool.

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>

Identifies the virtual machine by means of a VM index or VM name.

CPU-POOL-NAME = *BY-VM-GROUP(...)

Determines the CPU pool by specifying a VM group which is assigned to the CPU pool (S servers).

VM-GROUP-NAME = <name 1..8>

Name of the VM group.

INFORMATION =

This defines the output information.

INFORMATION = *CPU

Only information about CPUs contained in the CPU pool.

INFORMATION = *VM

Only information about the virtual machines assigned to the CPU pool.

INFORMATION = *SUMMARY

Summary of the information for *CPU and *VM.

INFORMATION = *ALL

All the information contained in the outputs *CPU and *VM.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
	64	VMS3002	Invalid operand value (CPU-IDENTIFICATION)
	64	VMS4000	VM not yet created
	64	VMS4043	CPU not known in system
	64	VMS4400	VM group not yet created
	64	VMS4600	CPU pool not yet created
	64	VMS4628	Spare CPUs do not belong to any CPU pool
	64	VMS9004	Functionality not supported on current HSI or with current parameter assignment

Output in S variables (in user task)

The `INFORMATION` operand of the command defines which S variables are to be supplied with values. The following can be specified for `INFORMATION`:

Notation of command	Abbreviated form used in table
<code>INFORMATION = *CPU</code>	1
<code>INFORMATION = *VM</code>	2
<code>INFORMATION = *SUMMARY</code>	3
<code>INFORMATION = *ALL</code>	4

Output information	Name of the S variable	T	Contents	Condition
Name of the CPU pool	<code>var(*LIST).CPU-POOL</code>	S	<code>*STDPPOOL / *POOL0</code> <code>*FOREIGN /</code> <name 1..8>	1,2,3,4
Scheduling procedures	<code>var(*LIST).SCHEDULE</code>	S	<code>*NONE</code> <code>*BY-TIMESLICE</code> <code>*DEDICATED-CPUS</code>	1,2,3,4
Number of assigned real normal CPUs	<code>var(*LIST).NUM-OF-NORMAL-CPUS.ALL</code>	I	<integer 0..64>	1,3,4
Number of assigned real extra CPUs	<code>var(*LIST).NUM-OF-EXTRA-CPUS.ALL</code>	I	<integer 0..63>	1,3,4
Real CPUs in the CPU pool	<code>var(*LIST).REAL-CPU(*LIST).ID</code>	S	<x-text 2..2>	1.4
Status of the real CPUs in the CPU pool	<code>var(*LIST).REAL-CPU(*LIST).STA</code>	S	<code>*ATT / *DET /</code> <code>*ERR / *OFFLINE</code> <code>*WAIT-FOR-MAINT</code>	1.4
Type of real CPUs in the CPU pool	<code>var(*LIST).REAL-CPU(*LIST).ATTR</code>	S	<code>*NORMAL / *EXTRA /</code> <code>*FOREIGN</code>	1.4
Number of assigned real foreign CPUs	<code>var(*LIST).NUM-OF-FOREIGN-CPUS.ALL</code>	I	<integer 0..63>	1.4
Number of VMs assigned to the CPU pool	<code>var(*LIST).NUM-OF-VMS</code>	I	<integer 0..99>	2,3,4
Index of the VM which is assigned to the CPU pool	<code>var(*LIST).VM(*LIST).VM-INDEX</code>	I	<integer 1..99>	2.4
Name of the VM which is assigned to the CPU pool	<code>var(*LIST).VM(*LIST).VM-NAME</code>	S	<name 1..8>	2.4
Name of the VM group which is assigned to the CPU pool	<code>var(*LIST).VM(*LIST).VM-GROUP</code>	S	<code>*NONE</code> <name 1..8>	2.4
Number of attached virtual CPUs of the VM which is assigned to the CPU pool	<code>var(*LIST).VM(*LIST).NUM-OF-ATT-CPUS</code>	I	<integer 0..32>	2.4
Multiprocessor level of the VM which is assigned to the CPU pool	<code>var(*LIST).VM(*LIST).NUM-OF-CPUS</code>	I	<integer 1..32>	2.4

Output information	Name of the S variable	T	Contents	Condition
Number of spare CPUs of the VM which is assigned to the CPU pool	var(*LIST).VM(*LIST).NUM-OF-SPARE-CPUS	I	<integer 0..15>	2.4
Number of assigned real normal CPUs (attached)	var(*LIST).NUM-OF-NORMAL-CPUS.ATT	I	<integer 0..64>	3
Number of assigned real normal CPUs (detached)	var(*LIST).NUM-OF-NORMAL-CPUS.DET	I	<integer 0..63>	3
Number of assigned real normal CPUs (offline)	var(*LIST).NUM-OF-NORMAL-CPUS.OFFLINE	I	<integer 0..63>	3
Number of assigned real extra CPUs (attached)	var(*LIST).NUM-OF-EXTRA-CPUS.ATT	I	<integer 0..63>	3
Number of assigned real extra CPUs (detached)	var(*LIST).NUM-OF-EXTRA-CPUS.DET	I	<integer 0..63>	3
Number of assigned real extra CPUs (offline)	var(*LIST).NUM-OF-EXTRA-CPUS.OFFLINE	I	<integer 0..63>	3
Total number of active virtual CPUs of the VMs assigned to the CPU pool	var(*LIST).NUM-OF-RUN-VIRT-CPUS	I	<integer 0..3168>	3

Output format for INFORMATION = *CPU*Example*

```

CPU-POOL SCHED CPU-ATTR REAL CPUS
*STDPOOL DC NORMAL 00(ATT),01(ATT)
          EXTRA 04(DET)
PCUST1   TS NORMAL 02(ATT),03(ATT)
          EXTRA NONE

```

Meaning of the output columns

CPU-POOL

Identifies the CPU pool by means of its name.

*STDPOOL Standard CPU pool

*POOL0 CPU pool of domain 0 (SQ servers)

*FOREIGN Foreign and application pool of the XenVMs (SQ servers)

SCHED

Scheduling procedure:

- There are no virtual CPUs in the scheduling procedure

DC Fixed CPU assignment (dedicated CPUs)

TS Time slicing

CPU-ATTR

Type of CPU (NORMAL / EXTRA / FOREIGN).

REAL CPUS

CPU identification and status in the format <cpu-id>(<rstate>),....

<cpu-id> CPU identifier.

<rstate> Status of the real CPU, see [page 214](#)
(ATT / DET / WFM / ERR / OFF).

NONE No CPU of this type belongs to the CPU pool.

Output format for INFORMATION = *VM*Example*

CPU-POOL	SCHED	VM-ID	VM-GROUP	ATT	PROC
PCUST1	TS	2 C1SYS1		2	4(+1)
		3 C1SYS2		2	2(+1)

Meaning of the output columns

CPU-POOL
SCHED

Corresponds to INFORMATION=*CPU.

VM-ID

Identifies a VM which is assigned to the CPU pool by means of VM index and VM name.
NONE No VM has been assigned to the CPU pool.

VM-GROUP

Name of the VM group if the VM is assigned to a VM group.
┆ The VM is not assigned to a VM group.

ATT

Number of attached virtual CPUs of the VM which is assigned to the CPU pool.

PROC

Virtual CPUs of the VM which is assigned to the CPU pool in the form m or $m(+n)$.
 m Number of virtual normal CPUs of the VM (multiprocessor level of the VM)
 n Number of virtual spare CPUs of the VM.

Output format for INFORMATION = *SUMMARY*Example*

```

CPU-POOL SCHED REAL-NORMAL-CPUS REAL-EXTRA-CPUS VIRT-CPUS
          ALL ATT DET OFF  ALL ATT DET OFF  RUN
PCUST1   TS    2  2  0  0    0  0  0  0    4

```

Meaning of the output columns

```

CPU-POOL
SCHED

```

Corresponds to INFORMATION=*CPU.

```

REAL-NORMAL-CPUS
ALL ATT DET OFF

```

Number of all / attached / detached / offline real normal CPUs of the CPU pool.

```

REAL-EXTRA-CPUS
ALL ATT DET OFF

```

Number of all / attached / detached / offline real extra CPUs of the CPU pool.

```

VIRT-CPUS
RUN

```

Total number of active virtual CPUs in the CPU pool.

Example

```

/show-vm-cpu-pool cpu-pool-name=pcust1,information=*all

```

```

CPU-POOL SCHED CPU-ATTR REAL CPUS
PCUST1   TS  NORMAL  02(ATT),03(ATT)
          EXTRA  NONE

```

```

CPU-POOL SCHED VM-ID      VM-GROUP ATT PROC
PCUST1   TS    2 C1SYS1          2 4(+1)
          3 C1SYS2          2 2(+1)

```

See also the detailed [“Example of managing CPU pools” on page 233](#).

SHOW-VM-DEVICE-STATUS

Output detailed device information

/SHOW-VM-DEVICE-STATUS outputs detailed information on VM2000 devices. This command can only be used by the VM2000 administrator and VM administrators with ADMIN-PRIVILEGES=*MAXIMUM.

The command provides the following information:

- the mnemonic device name
- the hardware attributes of the device (device type, PAV and SNAP attributes)
- the pubres attributes of the device (PUBRES-FOR-ATTRIBUTES / PUBRES-FORASSIGNMENT)
- the specifications for implicit assignment of the device (ASSIGN-BY-GUEST attribute, assignment set)
- the assignment of the device to a VM
- the assignment mode and the usage mode of the device
- the device assignments on the HA cluster (SQ servers)

(part 1 of 2)

SHOW-VM-DEVICE-STATUS

```

UNITS = *STD / *ALL / *PUBSET-DEVICES(...) /
      list-poss(28): *DEVICE-RANGE(...) / <alphanum-name 1..4 with-wild(20)>

*PUBSET-DEVICES(...)
  |
  | PUBSET = *BY-PUBRES-DEVICE(...) / <cat-id 1..4>
  |   *BY-PUBRES-DEVICE(...)
  |     |
  |     | UNIT = <alphanum-name 2..2> / <x-text 4..4>
  |
  |
  | *DEVICE-RANGE(...)
  |   |
  |   | FROM = <alphanum-name 2..2> / <x-text 4..4>
  |   | ,TO = <alphanum-name 2..2> / <x-text 4..4>
  |
  |

```

```

,SELECT = *PARAMETERS(...)
  *PARAMETERS(...)
    |
    | TYPE = *ANY / *VC / <family> / <device>
    | ,STATE = *ANY / *ASSIGNED(...) / *DISPOSABLE(...)
    |   *ASSIGNED(...)
    |     |
    |     | BY = *ANY / *VM-ADMIN / *GUEST
    |     | ,USAGE = *ANY / *EXCLUSIVE / *SHARED
    |     | ,VM-IDENTIFICATION = *ANY / <integer 1..99> / <name 1..8 with-wild(20)> /
    |     |       *BY-VM-GROUP(...) / *BY-CPU-POOL(...)
    |     |   *BY-VM-GROUP(...)
    |     |     |
    |     |     | VM-GROUP-NAME = <name 1..8>
    |     |   *BY-CPU-POOL(...)
    |     |     |
    |     |     | CPU-POOL-NAME = *STDPOOL / <name 1..8>
    |   *DISPOSABLE(...)
    |     |
    |     | CURRENT-STATE = *ANY / *UNASSIGNED / *SHARED / *KEEP
    | ,ASSIGN-BY-GUEST = *ANY / *NO / *YES(...)
    |   *YES(...)
    |     |
    |     | SET-NAME = *ANY / *STDSET / *BY-GUEST(...) / <name 1..8>
    |     |   *BY-GUEST(...)
    |     |     |
    |     |     | GUEST = <integer 1..99> / <name 1..8>
    | ,SPECIFIED-AS-PUBRES = *ANY / *NO / *YES(...)
    |   *YES(...)
    |     |
    |     | FOR = *ANY / *ASSIGNMENT / *ATTRIBUTES
    | ,SNAP-DEVICE = *ANY / *NO / *YES
    | ,PAV-ATTRIBUTE = *ANY / *PAV / *NO-PAV / *BASE / *ALIAS / *NO-ALIAS
,INFORMATION = *STD / *ALL / *VM-SUMMARY / *PAV / *HA-CLUSTER(...)
  *HA-CLUSTER(...)
    |
    | CONFLICT-DEVICE = *ANY | *NO | *YES

```

UNITS =

Specifies the devices whose device information is to be output.

UNITS = *STD

Information on all devices is output. If this operand is entered on a BS2000 console or in the ADMIN dialog via \$VMCONS, the information to be output must be limited by specifying SELECT=... or INFORMATION=*VMSUMMARY/*HA-CLUSTER.

UNITS = *ALL

Information on all devices is output. The output set can be extensive unless other entries are made to limit the information to be output.

UNITS = *PUBSET-DEVICES(...)

Information is output on all devices of a pubset.

PUBSET = *BY-PUBRES-DEVICE(...)

Determines the devices of the pubset by means of the pubres device.

UNIT = <alphanum-name 2..2> / <x-text 4..4>

Mnemonic device name of the pubres device.

PUBSET = <cat-id 1..4>

Catalog ID of the pubset. The pubset must be entered together with its pubres device in the monitor system's MRS catalog.

UNITS = *DEVICE-RANGE(...)

Information is to be output for all the devices in this name range.



Device names <x-text 4..4> (in *DEVICE-RANGE) are selected from FROM to TO in hexadecimal order. Otherwise, device names <alphanum-name 2..2> are selected in alphabetical order.

FROM = <alphanum-name 2..2> / <x-text 4..4>

Defines the start of the name range.

TO = <alphanum-name 2..2> / <x-text 4..4>

Defines the end of the name range. The mnemonic device name must be in the same value range (2- or 4-digit mn) as the FROM device name and must be greater than or equal to the FROM device name in ascending order in the value range.

UNITS = <alphanum-name 1..4 with-wild(20)>

Specifies the devices whose device information is to be output.

SELECT = *PARAMETERS(...)

Selects from the (possibly large) set of devices in the UNITS operand a suitable subset about which detailed device information is to be output.

TYPE =

Selects devices on the basis of the device type.

TYPE = *ANY

No selection is made on the basis of the device type.

TYPE = *VC

Selects devices which are assigned as virtual console (STATE=*ASSIGNED) or can be assigned as virtual console (STATE=*DISPOSABLE).

TYPE = <family>

Outputs all the devices of the specified device family.

<family>	<family> name
*CONSOLE	Console
*PRINTER	Printer
*FAM50	Special devices
*TD	Data transmission device
*EXOTIC	Physically supported device (ADAM)
*UR	Unit record device
*DISK	Disk device
*UNMTAPE	Unimodal tape device
*MBK	Magnetic tape cartridge device
*BIMTAPE	Bimodal tape device
*TAPE	Tape device

TYPE = <device>

Outputs all the devices of the specified device type. The possible device types are contained in the device type table (Device type column) in the "System Installation" manual [1]. The possible device and volume types are displayed in the dialog with TYPE=?. The device types on the current system can be displayed by means of /SHOW-DEVICE-CONFIGURATION, for example.

STATE =

Selects devices on the basis of their assignment status.

STATE = *ANY

No selection is made on the basis of the assignment status.

STATE = *ASSIGNED(...)

Outputs all devices that are assigned to a VM. The VM is selected by means of the `VMIDENTIFICATION` operand. The `BY` and `USAGE` operands define the set of devices to be output more closely.

BY =

Selects devices on the basis of who has assigned the device to at least one VM.

BY = *ANY

No selection is made on this basis.

BY = *VM-ADMIN

Outputs all devices assigned explicitly by the VM or VM2000 administrator. If a VM was explicitly selected in the `VM-IDENTIFICATION` operand, this selection attribute must apply for this VM.

BY = *GUEST

Outputs all devices assigned implicitly to the VM by the guest system. If a VM was explicitly selected in the `VM-IDENTIFICATION` operand, this selection attribute must apply for this VM.

USAGE =

Selects devices on the basis of the usage mode.

USAGE = *ANY

No selection is made on the basis of the usage mode.

USAGE = *EXCLUSIVE

Outputs all devices that are assigned to a VM for exclusive use.

USAGE = *SHARED

Outputs all devices that are assigned to a VM for shared use with other VMs.

VM-IDENTIFICATION =

Selects devices on the basis of the VM to which they are assigned.

VM-IDENTIFICATION = *ANY

No selection is made on this basis.

VM-IDENTIFICATION = <integer 1..99> / <name 1..8 with-wild(20)>

Identifies the virtual machine by means of a VM index or VM name.

VM-IDENTIFICATION = *BY-VM-GROUP(...)

Information is output for all VMs which belong to the VM group (S servers).

VM-GROUP-NAME = <name 1..8>

Name of the VM group.

VM-IDENTIFICATION = *BY-CPU-POOL(...)

Information is output for all VMs which belong to the CPU pool.

CPU-POOL-NAME = *STDPOOL / <name 1..8>

Name of the CPU pool.

STATE = *DISPOSABLE(...)

Outputs all devices that can currently be assigned.

CURRENT-STATE =

Selects devices on the basis of their assignment status.

CURRENT-STATE = *ANY

No selection is made on the basis of the assignment status.

CURRENT-STATE = *UNASSIGNED

Outputs all devices not assigned to a VM (free devices).

CURRENT-STATE = *SHARED

Outputs all devices that are assigned for shared use with other VMs. These devices are assigned to at least one VM.

CURRENT-STATE = *KEEP

Outputs all MTC devices in the `KEEP` state (see [page 290](#)).

ASSIGN-BY-GUEST =

Selects devices on the basis of whether they have to be assigned explicitly by the VM or VM2000 administrator or can be assigned to a suitable VM implicitly by the guest system.

ASSIGN-BY-GUEST = *ANY

No selection is made on this basis.

ASSIGN-BY-GUEST = *NO

Outputs all devices that can only be assigned by the VM or VM2000 administrator.
Exception: Alias devices (see [page 347](#)) and snap units for Snapsets (see [page 336](#)).

ASSIGN-BY-GUEST = *YES(...)

Outputs all devices that can be assigned to a suitable VM implicitly by the guest system. Alias devices (see [page 347](#)) and snap units for Snapsets (see [page 336](#)) are not taken into consideration.

SET-NAME = *ANY / *STDSET / <name 1..8>

Outputs all devices which are assigned to any or to the specified assignment set. With `INFORMATION=*HA-CLUSTER` this name is evaluated only for assignment sets of the local Server Unit.

SET-NAME = *BY-GUEST(...)

Outputs all devices which the guest system on the VM may implicitly assign to itself.

GUEST = <integer 1..99 / <name 1..8>

Selects the VM by means of the VM index or VM name.

SPECIFIED-AS-PUBRES =

Selects the devices on the basis of whether they are known as pubres devices.

SPECIFIED-AS-PUBRES = *ANY

No selection is made on this basis.

SPECIFIED-AS-PUBRES = *NO

Outputs all devices which are not known as pubres devices.

SPECIFIED-AS-PUBRES = *YES(...)

Selects the pubres devices on the basis of their pubres attribute.

FOR = *ANY

No selection is made on this basis.

FOR = *ASSIGNMENT

Outputs all pubres devices which have the `PUBRES-FOR-ASSIGNMENT` attribute for an arbitrary VM. If a VM was explicitly selected in the `VM-IDENTIFICATION` operand, this selection attribute must apply for this VM.

FOR = *ATTRIBUTES

Outputs all pubres devices which have the `PUBRES-FOR-ASSIGNMENT` attribute.

SNAP-DEVICE =

Selects the devices on the basis of whether they are configured as snap units (see [page 336](#)).

SNAP-DEVICE = *ANY

No selection.

SNAP-DEVICE = *NO

Outputs all devices which are not configured as snap units.

SNAP-DEVICE = *YES

Outputs all devices which are configured as snap units.

PAV-ATTRIBUTE =

Selects devices on the basis of whether they are configured as PAV (base device or alias device). See [page 345](#).

PAV-ATTRIBUTE = *ANY

No selection.

PAV-ATTRIBUTE = *PAV

Outputs all detected base or alias devices.

PAV-ATTRIBUTE = *NO-PAV

Outputs all devices that are not base or alias devices.

PAV-ATTRIBUTE = *BASE

Outputs all detected base devices.

PAV-ATTRIBUTE = *ALIAS

Outputs all detected alias devices.

PAV-ATTRIBUTE = *NO-ALIAS

Outputs all devices that are not alias devices.

INFORMATION =

Specifies the output information.

INFORMATION = *STD

Outputs detailed information on each device. If the device is assigned to several VMs, only the number of VMs is output.

INFORMATION = *ALL

Outputs detailed information on each device. If the device is assigned to several VMs, all VMs are output. In addition, the PAV property of the device may be output.

INFORMATION = *VM-SUMMARY

Compact output of the devices which are assigned to a VM, sorted by VMs. Devices are then output which can still be assigned to a VM.

INFORMATION = *PAV

Outputs detailed device information including the configuration of the PAV devices (base and alias devices).

INFORMATION = *HA-CLUSTER(...)

Compact output of the device assignments and devices which can still be assigned to a VM and which are sorted according to Server Units in the HA cluster and VMs (SQ servers).

CONFLICT-DEVICE =

Selects the devices on the basis of whether a conflict exists for the device assignments in the HA cluster.

CONFLICT-DEVICE = *ANY

No selection.

CONFLICT-DEVICE = *NO

Selects all devices for which no conflict exists.

CONFLICT-DEVICE = *YES

Selects all devices for which a conflict exists.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
2	0	CMD0001	Command successfully executed; not all devices available
2	0	VMS2200	VM2000 not running on a Server Unit in an HA cluster
2	0	VMS4001	No VM with appropriate name found
1	0	VMS4035	No corresponding device found (SELECT)
1	0	VMS4180	Specified set of devices empty
2	0	VMS4238	Only information on pubres device (*PUBSET-DEVICES)
2	0	VMS4403	No VM belongs to VM group
2	0	VMS4613	No VM has been assigned to CPU pool
	64	VMS3002	Invalid operand value (e.g. TYPE or *DEVICE-RANGE)
	64	VMS3037	UNITS=*STD, no additional selection and input from console or via \$VMCONS
	64	VMS4000	VM not yet created
	64	VMS4200	Assignment set not yet created
	64	VMS4223	Command not executed (operand *PUBSET-DEVICES)
	64	VMS4400	VM group not yet created
	64	VMS4600	CPU pool not yet created
	130	VMS2201	No information on the HA cluster available

Guaranteed messages

Only if UNITS is specified without *DEVICE-RANGE and without wildcards: VMS4006

Output in S variables (in user task)

The `INFORMATION` operand of the command defines which S variables are to be supplied with values. The following can be specified for `INFORMATION`:

Notation of command	Abbreviated form used in table
<code>INFORMATION = *STD</code>	1
<code>INFORMATION = *ALL</code>	2
<code>INFORMATION = *PAV</code>	3
<code>INFORMATION = *VM-SUMMARY</code>	4
<code>INFORMATION = *HA-CLUSTER</code>	5

Output information	Name of the S variable	T	Contents	Condition
Mnemonic device name	<code>var(*LIST).MN</code>	S	<alphanum-name 2..2> <x-text 4..4>	1,2,3
Device type	<code>var(*LIST).TYPE</code>	S	<name 1..8>	1,2,3
Can the device be used as virtual console?	<code>var(*LIST).VC</code>	B	TRUE / FALSE	1,2,3
Is the device configured as a snap unit?	<code>var(*LIST).SNAP</code>	B	TRUE / FALSE	1,2,3
Can the device be assigned implicitly by the guest system?	<code>var(*LIST).ASS-BY-GUEST</code>	B	TRUE / FALSE	1,2,3
Assignment set to which the device is assigned	<code>var(*LIST).ASSIGNMENT-SET</code>	S	<name 1..8> *NONE / *STDSET	1,2,3
Does the device have the attribute <code>PUBRES-FOR-ATTRIBUTES</code> ?	<code>var(*LIST).PUBRES-FOR-ATTR</code>	B	TRUE / FALSE	1,2,3
PAV property of the device	<code>var(*LIST).PAV-ATTR</code>	S	*NO-PAV / *BASE *ALIAS	1,2,3
Base device of the alias device (PAV)	<code>var(*LIST).BASE-MN</code>	S	<alphanum-name 2..2> <x-text 4..4> *NONE	1,2,3
Alias devices of the base device (PAV)	<code>var(*LIST).ALIAS(*LIST).ALIAS-MN</code>	S	<alphanum-name 2..2> <x-text 4..4>	1,2,3
Usage mode	<code>var(*LIST).USAGE</code>	S	*EXCL / *SHARED *UNDEF	1,2,3
Type of I/O processing	<code>var(*LIST).IO-MODE</code>	S	*UNDEF / *DIRECT *INDIRECT	1,2,3
Index of the VM to which the device is assigned	<code>var(*LIST).ASS(*LIST).TO-VM-INDEX</code>	I	<integer 1..99>	1,2,3
Name of the VM to which the device is assigned	<code>var(*LIST).ASS(*LIST).TO-VM-NAME</code>	S	<name 1..8>	1,2,3
Who assigned the device to the VM?	<code>var(*LIST).ASS(*LIST).BY</code>	S	*VM-ADMIN *GUEST	1,2,3

Output information	Name of the S variable	T	Contents	Condition
Was the device assigned to the VM as a pubres device?	var(*LIST).ASS(*LIST).PUBRES-FOR-ASSIGN	B	TRUE / FALSE	1,2,3
Is the device in KEEP mode?	var(*LIST).ASS(*LIST).KEEP	B	TRUE / FALSE	1,2,3
Index of VM	var(*LIST).VM-INDEX	I	<integer 0..99> ¹	4.5
Name of VM	var(*LIST).VM-NAME	S	<name 1..8> *NONE ¹	4.5
Server Unit in the HA cluster on which the VM runs (0: local Server Unit, 1: remote Server Unit)	var(*LIST).SERVER-UNIT-INDEX	I	<integer 0..1>	5
Is the VM highly available?	var(*LIST).HA	B	TRUE / FALSE	5
Mnemonic device name	var(*LIST).UNIT(*LIST).MN	S	<alphanum-name 2..2> <x-text 4..4>	4.5
Who assigned the device to the VM?	var(*LIST).UNIT(*LIST).ASS-BY	S	*VM-ADMIN *GUEST / *UNDEF	4.5
Usage mode	var(*LIST).UNIT(*LIST).USAGE	S	*EXCL / *SHARED *UNDEF	4.5
Is the device in KEEP mode?	var(*LIST).UNIT(*LIST).KEEP	B	TRUE / FALSE	4.5
Is the device a virtual console?	var(*LIST).UNIT(*LIST).VC	B	TRUE / FALSE	4.5
Is the device on the local Server Unit not configured in the HA cluster?	var(*LIST).UNIT(*LIST).NOT-GENERATED	B	TRUE / FALSE	5
Conflict in the HA cluster because of the device's assignments?	var(*LIST).UNIT(*LIST).ASS-CONFL	B	TRUE / FALSE	5
Possible usage mode when assigned to a non-highly available VM	var(*LIST).UNIT(*LIST).DISPO-NHA	S	*EXCL / *SHARED / *UNDEF	5
Possible usage mode when assigned to a highly available VM	var(*LIST).UNIT(*LIST).DISPO-HA	S	*EXCL / *SHARED / *UNDEF	5

¹ 0 or *NONE as VM identification for devices that are not assigned to a VM

Output format for INFORMATION = *STD / *ALL*Example (INFORMATION=*ALL)*

UNIT	DEV-TYPE	PAV	S	ASSIGN-SET	USAGE	VM-ID/(#VM)	ASSIGNED-BY
D3	D3435			SYS2IMPL	SH(I)	(2)	MIXED
						1	MONITOR VM-ADMIN
						2	SYSTEM2 GUEST
D4	D3435			SYS2IMPL(P)	SH(D)	2	SYSTEM2 GUEST
D5	D3435			SYS2IMPL	EXCL	2	SYSTEM2 VM-ADMIN(P)
X2	HNC			SYS2IMPL	EXCL	2	SYSTEM2 GUEST
X3	HNC			SYS2IMPL	EXCL	2	SYSTEM2 GUEST

Meaning of the output columns

UNIT

Mnemonic device name.

DEV-TYPE

Device type (*VC for virtual consoles).

PAV

PAV attribute of the device.BASE **The device is a base device.**ALIAS **The device is an alias device.**

S

SNAP attribute of the device.S **The device is a snap unit.**

ASSIGN-SET

Assignment set of the device.

*STDSET / <name 1..8>

Name of the assignment set._ **The device cannot be assigned implicitly.**...(P) **This suffix indicates a pubres device with the attribute PUBRES-FOR-ATTRIBUTES (see [page 305](#)).**

USAGE

Usage mode of the device.EXCL **The device is used exclusively by a VM.**SH(D) **The device is only assigned to one VM. Inputs/outputs are processed directly. The device can be assigned to other VMs for shared use.**SH(I) **The device is currently assigned to more than one VM.****[S]** **The inputs/outputs are processed indirectly.**

VM-ID/(#VM)

Identification of the VM (<vm-index>_<vm-name>) when the device is assigned in the EXCL or SH(D) usage mode, or (n, n≥1) number of VMs that share the device (SH(I) usage mode).

For INFORMATION=*ALL, output lines follow that (only) in this output column and the ASSIGNED-BY column show the identification of the VMs that share the device and the type of device assignment.

ASSIGNED-BY

Specifies who the device was assigned to the VM by:

VM-ADMIN	Device assigned explicitly by the VM or VM2000 administrator.
VM-ADMIN(P)	Device assigned explicitly as pubres device by the VM or VM2000 administrator. The pubres device therefore has the PUBRES-FOR-ASSIGNMENT attribute for this VM (see page 305).
GUEST	Device assigned to the VM implicitly by the guest system.
GUEST(KEEP)	After detachment the implicit assignment for an MTC device is preserved (see page 290).
MIXED	Device assigned to multiple VMs as a shareable device in various procedures. This display appears only in the output line with the number of VMs which share this device.

Output format for INFORMATION = *PAV

Example

```
UNIT DEV-TYPE PAV S ASSIGN-SET USAGE VM-ID/(#VM) ASSIGNED-BY
2300 D3435 BASE EXCL 1 MONITOR VM-ADMIN
(BASE: 2300; ALIAS: 2310,2320,2330)
```

Meaning of the output columns

The meaning of the output columns corresponds to INFORMATION=*STD.

For each base or alias device, the output line is followed by an information line that indicates the structure of the PAV (i.e. base device and associated alias devices):

(BASE: <mn>; ALIAS: <mn>,...)

Output format for INFORMATION = *VM-SUMMARY*Example*

```

VM-ID          UNITS
1 MONITOR     ASSIGNED BY VM-ADMIN
                EXCL:  C2,C3,V0,V1,2430,2431,2432,2433
                SHARED: 22F4,22F5,22F6,22F7,58F2
                ASSIGNED BY GUEST
                SHARED: E062
2 SYSTEM2     ASSIGNED BY VM-ADMIN
                EXCL:  CH,CI,E05F,E060,E061
                SHARED: E062
DISPOSABLE    UNASS:  CB,CC,CD,CE,CF,CG,CJ,CK,...
                SHARED: 22F4,22F5,22F6,22F7,58F2,E062,...

```

Meaning of the output columns

VM-ID

Identifies the VM by means of VM index and VM name.

DISPOSABLE The devices can be assigned to a VM.

UNITS

The output of the devices assigned to a VM takes place VM-specifically in blocks: first the explicitly assigned devices (ASSIGNED BY VM-ADMIN), then the implicitly assigned devices (ASSIGNED BY GUEST). The output is grouped in accordance with the usage mode (EXCL/SHARED/KEEP). Empty blocks or groups are omitted.

The output of the devices that can be assigned to a VM (DISPOSABLE) is grouped by its type (UNASS/SHARED/KEEP). See the /SHOW-VM-DEVICE-STATUS operand STATE=*DISPOSABLE. Empty groups are omitted.

The device list is arranged in ascending order of the value range (two- or four-digit). The two-digit device names are output first, followed by the four-digit device names.

- Ⓢ For S servers the suffix (VC) identifies a device that is used or can be used as a virtual console (see [page 117](#)).

Output format for INFORMATION = *HA-CLUSTER (SQ servers)*Example*

```

SU VM-ID      HA UNITS
L   1 MONITOR  N  ASSIGNED BY VM-ADMIN
                        EXCL:  X0,X1,Y0,Y1,Z0,Z1,9904,9905,9906,990A,990B,990C
L   2 SHIRKVM2 Y  ASSIGNED BY VM-ADMIN
                        EXCL:  AB(!),D2,X2,X3,Y2,Y3,Z2,Z3
                        SHARED: D0
                        ASSIGNED BY GUEST
                        EXCL:  AD,AF
L   3 SHIRKVM3 N  ASSIGNED BY VM-ADMIN
                        EXCL:  D3,X4,X5,Y4,Y5,Z4,Z5
                        SHARED: AA(!),AE
R   1 MONITOR  N  ASSIGNED BY VM-ADMIN
                        EXCL:  D5,X0,X1,Y0,Y1,Z0,Z1
R   2 PRODVM   Y  ASSIGNED BY VM-ADMIN
                        EXCL:  AA(!),D4,X6,X7,Y6,Y7,Z6,Z7,990D,990E
                        SHARED: D0,9900,9901
                        ASSIGNED BY GUEST
                        EXCL:  AE
R   2 TESTVM   N  ASSIGNED BY VM-ADMIN
                        EXCL:  AF,X4,X5,Y4,Y5,Z4,Z5,9907,9908,9909
                        SHARED: D0
                        ASSIGNED BY GUEST
                        EXCL:  AB(!),AC,AD,9999(?)
                        SHARED: 990F
L   DISPOSABLE N  UNASS: AC,D5,9907,9908,9909,990F
L   DISPOSABLE YN UNASS: AG,AH,...,D1,...,X8,X9,Y8,Y9,Z8,Z9,...,9902,9903
                        SHARED: D0,9900,9901,990F

```

Meaning of the output columns

SU

Identifies the Server Unit in the HA cluster

L Local Server Unit in the HA cluster.

R Remote Server Unit in the HA cluster.

VM-ID

Identifies the VM by means of VM index and VM name.

DISPOSABLE The devices can be assigned to a VM.

Output of "-" for the VM index means that the highly available VM is currently not running on this Server Unit (there is only a VM configuration file).

HA

High-availability attribute (for a VM).

Y The VM is highly available.

N The VM is not highly available.

High-availability attribute (for devices which can be assigned to a VM, DISPOSABLE). A device can be represented in both output blocks, in the preceding example: 990F.

YN Device available for VMs which are highly available or not highly available.

N Device available only for VMs which are not highly available

UNITS

The output of the devices assigned to a VM takes place VM-specifically in blocks: first the explicitly assigned devices (ASSIGNED BY VM-ADMIN), then the implicitly assigned devices (ASSIGNED BY GUEST). The output is grouped in accordance with the usage mode (EXCL/SHARED/KEEP). Empty blocks or groups are omitted.

The output of the devices that can be assigned to a VM (DISPOSABLE) is grouped according to its high-availability attribute (N/YN) and according to its assignment status (UNASS/SHARED/KEEP), see the /SHOW-VM-DEVICE-STATUS operand STATE=*DISPOSABLE. Empty groups are omitted.

The device list is arranged in ascending order of the value range (two- or four-digit). The two-digit device names are output first, followed by the four-digit device names.



The output (!) indicates a possible device conflict in the HA cluster.

It was not possible to assign the device to a highly available VM in the event of a failover on the other Server Unit, in the example above: AA and AB.

The output (?) marks a device which is assigned to a VM on the remote Server Unit in the HA cluster but which is not configured on the local Server Unit in the HA cluster, in the example above: 9999.

Example

```

/show-vm-device-status units=*std,information=*all,
    select=*par(assign-by-guest=*yes(set-name=sys2impl)) ----- (1)
UNIT DEV-TYPE PAV S ASSIGN-SET USAGE VM-ID/(#VM) ASSIGNED-BY
D3 D3435 SYS2IMPL SH(I) ( 2) MIXED ----- (2)
    1 MONITOR VM-ADMIN
    2 SYSTEM2 GUEST
D4 D3435 SYS2IMPL(P) SH(D) 2 SYSTEM2 GUEST ----- (3)
D5 D3435 SYS2IMPL EXCL 2 SYSTEM2 VM-ADMIN(P)
X2 HNC SYS2IMPL EXCL 2 SYSTEM2 GUEST ----- (4)
X3 HNC SYS2IMPL EXCL 2 SYSTEM2 GUEST
/show-vm-device-status units=*std,information=*pav,
    select=*par(state=*disposable,pav-attribute=*base) ----- (5)
UNIT DEV-TYPE PAV S ASSIGN-SET USAGE VM-ID/(#VM) ASSIGNED-BY
2300 STDDISK BASE
    (BASE: 2300; ALIAS: 2310,2320,2330)
2301 STDDISK BASE
    (BASE: 2301; ALIAS: 2311,2321,2331)
2302 STDDISK BASE
    (BASE: 2302; ALIAS: 2312,2322,2332)
... (Ausgabe gekürzt)
5400 STDDISK BASE
    (BASE: 5400; ALIAS: 540C)

```

- (1) The VM2000 administrator obtains detailed information on the status of the devices of assignment set `SYS2IMPL`. If `INFORMATION=*ALL` is specified, all VMs are output to which a device is assigned in the `SH(I)` usage mode.
- (2) Disk `D3` is assigned to the VMs `MONITOR` and `SYSTEM2` for shared use. The usage mode is therefore `SH(I)` (SHARED with indirect I/O). `D3` has been assigned to `MONITOR` explicitly and to `SYSTEM2` implicitly.
- (3) In the case of implicit device assignment, the disk devices are assigned in the usage mode `SHARED`. Input/output takes place by means of direct I/O because the devices are initially assigned to only one VM.
- (4) In the case of implicit device assignment, other devices (see [page 289](#)) are assigned in the usage mode `EXCLUSIVE`.
- (5) The VM2000 administrator finds out which base devices (PAV) are free. Detailed information is to be displayed on the configuration of the PAV devices (base and alias devices).

See also the detailed examples for device and pubset management on [page 294](#) and on [page 324](#).

SHOW-VM-GROUP

Output attributes of VM group (S servers)

SQ No VM groups are available on SQ servers. The command is rejected with message VMS3011 and with general return code VMS9004.

/SHOW-VM-GROUP is used to output the attributes of VM groups.

This command is only permitted for the VM2000 administrator.

The command provides the following information:

- The attributes of the VM group as defined with /CREATE-VM-GROUP or /MODIFY-VM-GROUP-ATTRIBUTES
- the VMs which belong to the VM group and their attributes

SHOW-VM-GROUP
VM-GROUP-NAME = *ALL / <name 1..8> / *BY-MEMBER(...) / *BY-CPU-POOL(...) *BY-MEMBER(...) VM-IDENTIFICATION = <integer 1..99> / <name 1..8> *BY-CPU-POOL(...) CPU-POOL-NAME = *STDPOOL / <name 1..8> ,INFORMATION = *ATTRIBUTES / *MEMBERS / *ALL

VM-GROUP-NAME =

Determines the VM group about which information is to be output.

VM-GROUP-NAME = ***ALL**

Information about all VM groups is to be output.

VM-GROUP-NAME = <name 1..8>

Name of the VM group.

VM-GROUP-NAME = ***BY-MEMBER(...)**

Determines the VM group by specifying a VM belonging to the VM group.

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>

Identifies the virtual machine by means of a VM index or VM name.

VM-GROUP-NAME = *BY-CPU-POOL(...)

Determines the VM groups by specifying the CPU pool to which the VM groups are assigned.

CPU-POOL-NAME = *STDPOOL / <name 1..8>

Name of the CPU pool.

INFORMATION =

Specifies the output information.

INFORMATION = *ATTRIBUTES

Only attributes of the VM group.

INFORMATION = *MEMBERS

Only information about the members of the VM group.

INFORMATION = *ALL

All the information contained in the outputs *ATTRIBUTES and *MEMBERS.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
2	0	VMS4411	No VM group created (VM-GROUP-NAME=*ALL)
2	0	VMS4623	No VM group assigned to CPU pool
	64	VMS4000	VM not yet created
	64	VMS4400	VM group not yet created
	64	VMS4428	VM does not belong to a VM group
	64	VMS4600	CPU pool not yet created

Output in S variables (in user task)

The `INFORMATION` operand of the command defines which S variables are supplied with values. The following can be specified for `INFORMATION`:

Notation of command	Abbreviated form used in table
<code>INFORMATION = *ATTRIBUTES</code>	1
<code>INFORMATION = *MEMBERS</code>	2
<code>INFORMATION = *ALL</code>	3

Output information	Name of the S variable	T	Contents	Condition
Name of the VM group	<code>var(*LIST).VM-GROUP</code>	S	<name 1..8>	1,2,3
Name of the CPU pool	<code>var(*LIST).CPU-POOL</code>	S	*STDPOOL <name 1..8>	1,2,3
CPU quota of the VM group	<code>var(*LIST).CPU-QUOTA</code>	S	<fixed 0.01..99.99>	1,3
Maximum CPU utilization of the VM group	<code>var(*LIST).MAX-CPU-UTILIZATION</code>	S	<fixed 0.01..100.00>	1,3
Number of VMs which belong to the VM group	<code>var(*LIST).NUM-OF-MEMBERS</code>	I	<integer 0..99>	2,3
Index of VM	<code>var(*LIST).MEMBER(*LIST).VM-INDEX</code>	I	<integer 1..99>	2,3
Name of VM	<code>var(*LIST).MEMBER(*LIST).VM-NAME</code>	S	<name 1..8>	2,3
Share of the VM in the CPU quota of the VM group	<code>var(*LIST).MEMBER(*LIST).MEMBER-CPU-QUOTA</code>	S	<fixed 0.01..99.99>	2,3
Effective CPU share of the VM	<code>var(*LIST).MEMBER(*LIST).EFFECTIVE-CPU-QUOTA</code>	S	<fixed 0.01..100.00>	2,3
Maximum CPU utilization of the VM	<code>var(*LIST).MEMBER(*LIST).MAX-CPU-UTILIZATION</code>	S	<fixed 0.01..100.00>	2,3
Effective maximum CPU utilization of the VM	<code>var(*LIST).MEMBER(*LIST).EFFECTIVE-MAX-UTIL</code>	S	<fixed 0.01..100.00>	2,3

Output format for INFORMATION = *ATTRIBUTES*Example*

```
VM-GROUP  CPU-POOL CPU-Q   MAX-U
GCUST1    *STDPOOL  20.00  20.00
```

Meaning of the output columns

VM-GROUP

Identifies the VM group by means of its name.

CPU-POOL

CPU pool to which the VM group is assigned.

*STDPOOL The VM group is assigned to the standard CPU pool.

CPU-Q

CPU quota of the VM group as assigned by means of command (CPU-QUOTA operand).

MAX-U

Maximum CPU utilization of the VM group as assigned by means of command (MAX-CPU-UTILIZATION operand).

Output format for INFORMATION = *MEMBERS*Example*

```

VM-GROUP CPU-POOL VM-ID          CPU-Q / EFF-Q   MAX-U / EFF-U
GCUST1   *STDPOOL 2 C1SYS1      1.00 / 10.00   20.00 / 20.00
                   3 C1SYS2      1.00 / 10.00  100.00 / 20.00

```

Meaning of the output columns

VM-GROUP

Identifies the VM group by means of its name.

CPU-POOL

CPU pool to which the VM group is assigned.***STDPOOL** The VM group is assigned to the standard CPU pool.

VM-ID

Identifies a VM which belongs to the VM group by means of VM index and VM name.**NONE** No VM has been added to the VM group.

CPU-Q / EFF-Q

CPU quota of the VM in the format <member-cpu-q> / <effective-cpu-q>.**<member-cpu-q>** Member CPU quota of the VM, set by means of command (MEMBER-CPU-QUOTA operand).**<effective-cpu-q>** Effective CPU share of the VM, calculated by VM2000. Values < 0.01 are rounded up to 0.01.

MAX-U / EFF-U

Maximum CPU utilization of the VM in the format <max-cpu-util> / <effective-max-cpu-util>.**<max-cpu-util>** Maximum CPU utilization set by means of command (MAX-CPU-UTILIZATION operand) for the VM.**<effective-max-cpu-util>** Effective maximum CPU utilization of the VM, calculated by VM2000.

Example

```
/show-vm-group vm-group-name=*all,information=*all _____ (1)
```

```
VM-GROUP CPU-Q MAX-U
GCUST1 20.00 20.00 _____ (2)
```

```
VM-GROUP VM-ID CPU-Q / EFF-Q MAX-U / EFF-U
GCUST1 2 C1SYS1 1.00 / 10.00 20.00 / 20.00 _____ (3)
        3 C1SYS2 1.00 / 10.00 100.00 / 20.00
```

- (1) The VM2000 administrator obtains information on the existing VM groups.
- (2) A VM group GCUST1 exists with a CPU quota and a maximum CPU utilization of 20.1
- (3) The VM group GCUST1 has two members, C1SYS1 and C1SYS2, with the same CPU quota. The maximum CPU utilization of C1SYS2 was set by means of command to 100, but is reduced to 20 by the corresponding value for the group.

See also the detailed [“Example of operating VM groups”](#) on page 182.

SHOW-VM-RESOURCES

Output status of VM2000 resources

The status of VM2000 resources is output using `/SHOW-VM-RESOURCES`. This command can only be used by the VM2000 administrator and VM administrators with `ADMIN-PRIVILEGES=*MAXIMUM`.

The command provides the following information:

- the attributes of all VMs, as defined by means of command
- the status of all VMs
- the name and version of the operating system on the guest system
- whether a VM administrator is connected to the VM
- the privileges of the VMs
- the number of non-initialized VMs and the greatest possible main memory size that can be assigned when a VM is set up
- the VM group to which the VM belongs
- the CPU pool to which the VM is assigned
- the scheduling procedure for the VM
- the settings of the VM for control via the real CPU
- the status of the virtual CPUs of all VMs and of the real CPUs
- the assignment of the devices to the VMs and the free devices (those not assigned to a VM); selection is possible on the basis of device types and properties
- the global storage partitions assigned to the VMs
- the main memory distribution over the VMs and the free main memory
- the monitoring job variables assigned to the VMs
- an overview of the server configuration
- Overview of the configuration of the VMs in the HA cluster (SQ servers)
- the assignment of the Channel Subsystem Monitoring Facility (S servers, see [page 392](#))

SHOW-VM-RESOURCES

```

INFORMATION = *STD / *CPU / *UNITS / *GS / *MEMORY / *MONJV / *PRIVILEGES /
          *CONFIGURATION / *HA-CLUSTER / *ALL
, UNITS = *ALL / list-poss(28): *DEVICE-RANGE(...) / <alphanum-name 1..4 with-wild(20)>
  *DEVICE-RANGE(...)
    | FROM = <alphanum-name 2..2> / <x-text 4..4>
    | ,TO = <alphanum-name 2..2> / <x-text 4..4>
, TYPE = *ANY / *SD / *VC / <family> / <device>
, ATTRIBUTES = *ALL / *ASSIGNED / *FREE
, VM-IDENTIFICATION = *VM2000 / <integer 1..99> / <name 1..8 with-wild(20)> /
  *BY-VM-GROUP(...) / *BY-CPU-POOL(...)
  *BY-VM-GROUP(...)
    | VM-GROUP-NAME = <name 1..8>
  *BY-CPU-POOL(...)
    | CPU-POOL-NAME = *STDPOOL / <name 1..8>

```

INFORMATION =

Specifies the output information.

INFORMATION = *STD

Only the attributes and status of the VM.

INFORMATION = *CPU

Only information on the CPU status.

INFORMATION = *UNITS

Only the devices are specified.

INFORMATION = *GS

Only information on global storage assigned to the VM.

INFORMATION = *MEMORY

Only the distribution of main memory

INFORMATION = *MONJV

Only information on the monitor job variables.

INFORMATION = *PRIVILEGES

Only information on the privileges of the VM.

INFORMATION = *CONFIGURATION

Only the overview of the server configuration.

INFORMATION = *HA-CLUSTER

Only the overview of the configuration of the VMs in the HA cluster (SQ servers).

INFORMATION = *ALL

All the information contained in the outputs *STD, *CPU, *UNITS, *GS, *MEMORY, *MONJV and *PRIVILEGES, but not in *CONFIGURATION and *HA-CLUSTER.

UNITS =

Specifies the devices whose assignment is to be output. This operand is only evaluated when INFORMATION=*UNITS/*ALL is specified at the same time.

UNITS = *ALL

The assignment of **all** devices is output in list form. The TYPE and ATTRIBUTES operands are taken into account.

UNITS = *DEVICE-RANGE(...)

The assignment of all the devices in this name range is to be output in list form. The TYPE and ATTRIBUTES operands are taken into account.



Device names <x-text 4..4> (in the case of *DEVICE-RANGE) are selected from FROM to TO in hexadecimal order. Otherwise, device names <alphanum-name 2..2> are selected in alphabetical order.

FROM = <alphanum-name 2..2> / <x-text 4..4>

Defines the start of the name range.

TO = <alphanum-name 2..2> / <x-text 4..4>

Defines the end of the name range. The mnemonic device name must be in the same value range (2- or 4-digit mn) as the FROM device name and must be greater than or equal to the FROM device name in ascending order in the value range.

UNITS = <alphanum-name 1..4 with-wild(20)>

Specifies the devices whose assignment is to be output.

If wildcards are used, the output is in list form. Without wildcards, a corresponding message is output for each device specified (the operand ATTRIBUTES is ignored in the case of output to SYSOUT, but has a meaning for output into an S variable, see [page 584](#)).

TYPE =

Selects the devices on the basis of their device properties. This operand is only evaluated when INFORMATION=*UNITS/*ALL.

TYPE = *ANY

The devices are not selected according to device attributes.

TYPE = *SD

Outputs all devices which are declared as shareable.

TYPE = *VC

Outputs all devices which are assigned as virtual console (ATTRIBUTES=*ASSIGNED) or can be assigned as virtual console (ATTRIBUTES=*FREE).

TYPE = <family>

Outputs all the devices of the specified device family.

<family>	<family> name
*CONSOLE	Console
*PRINTER	Printer
*FAM50	Special devices
*TD	TRANSDATA device
*EXOTIC	Physically supported device (ADAM)
*UR	Unit record device
*DISK	Disk device
*UNMTAPE	Unimodal tape device
*MBK	Magnetic tape cartridge device
*BIMTAPE	Bimodal tape device
*TAPE	Tape device

TYPE = <device>

Outputs all the devices of the specified device type. The possible device types are contained in the device type table (Device type column) in the "System Installation" manual [1]. The possible device and volume types are displayed in the dialog with TYPE=?. The device types on the current system can be displayed by means of /SHOW-DEVICE-CONFIGURATION, for example.

ATTRIBUTES =

Selects devices or main memory on the basis of whether they are assigned to a VM or can be assigned to a VM. This operand is only evaluated for devices if

INFORMATION=*UNITS/*ALL and UNITS=*ALL/*DEVICE-RANGE/<alphanum-name> are specified at the same time with wildcards.



With output in S variables, the effect of this operand to main memory is extended, see [page 584](#).

ATTRIBUTES = *ALL

Information for all devices or the entire main memory is output.

ATTRIBUTES = *ASSIGNED

Only the main memory of the VM or those devices assigned to a VM are output.

ATTRIBUTES = *FREE

Only those devices or main memory not assigned to a VM (DISPOSABLE) are output.

VM-IDENTIFICATION =

Identification of the virtual machine.



The operand is not evaluated for INFORMATION=*CONFIGURATION/*HA-CLUSTER.

VM-IDENTIFICATION = *VM2000

The information is output for all configured VMs.

VM-IDENTIFICATION = <integer 1..99> / <name 1..8 with-wild(20)>

Identifies the virtual machine by means of a VM index or VM name.

VM-IDENTIFICATION = *BY-VM-GROUP(...)

Information is output for all VMs which belong to the VM group (S servers).

VM-GROUP-NAME = <name 1..8>

Name of the VM group.

VM-IDENTIFICATION = *BY-CPU-POOL(...)

No selection is made on this basis.

CPU-POOL-NAME = *STDPOOL / <name 1..8>

Name of the CPU pool.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
1	0	VMS4035	No corresponding device found (ATTRIBUTES or TYPE)
1	0	VMS4180	Specified set of devices empty (INFORMATION=*ALL/*UNITS)
2	0	VMS2200	VM2000 not running on a Server Unit in an HA cluster
2	0	VMS4001	No VM with appropriate name found
2	0	VMS4403	No VM belongs to VM group
2	0	VMS4613	No VM has been assigned to CPU pool
	64	VMS3002	Invalid operand value (e.g. TYPE or *DEVICE-RANGE)
	64	VMS3003	Invalid combination of operands
	64	VMS4000	VM not yet created
	64	VMS4400	VM group not yet created
	64	VMS4600	CPU pool not yet created
	130	VMS2201	No information on the HA cluster available

Guaranteed messages

Only if UNITS is specified without *DEVICE-RANGE and without wildcards: VMS4006

Output in S variables (in user task)

CPU quota of the VM as set by means of command. The following can be specified for INFORMATION:

Notation of command	Abbreviated form used in table
INFORMATION = *STD	1
INFORMATION = *CPU	2
INFORMATION = *GS	3
INFORMATION = *MONJV	4
INFORMATION = *PRIVILEGES	5
INFORMATION = *CONFIGURATION	6
INFORMATION = *UNITS, ATTRIBUTE=*ALL/*ASSIGNED	7
INFORMATION = *MEMORY, ATTRIBUTE=*ALL/*ASSIGNED	8
INFORMATION = *ALL, ATTRIBUTE=*ALL/*ASSIGNED	9
INFORMATION = *UNITS, ATTRIBUTE=*FREE ¹	10
INFORMATION = *MEMORY, ATTRIBUTE=*FREE ¹	11
INFORMATION = *ALL, ATTRIBUTE=*FREE ¹	12
INFORMATION=*HA-CLUSTER	13

¹ The meaning of ATTRIBUTE=*FREE, which in the case of output to SYSOUT is only relevant for devices, has been extended for output in S variables. If ATTRIBUTE=*FREE is specified, the described information about the free devices or main memory which follows the VM-specific information in SYSOUT output is output for INFORMATION=*UNITS/*MEMORY/*ALL.

Output information	Name of the S variable	T	Contents	Condition
Index of VM	var(*LIST).VM-INDEX	I	<integer 1..99>	1 - 6, 7 - 9
Name of VM	var(*LIST).VM-NAME	S	<name 1..8>	1 - 6, 7 - 9
multiprocessor level of the VM	var(*LIST).NUM-OF-CPU	I	<integer 1..32>	1,2,9
Number of virtual spare CPUs of the VM	var(*LIST).NUM-OF-VM-SPARE-CPU	I	<integer 0..15>	1,2,9
Name of the CPU pool	var(*LIST).CPU-POOL	S	*STDPOOL <name 1..8>	1,2,9
CPU quota of the VM (integer)	var(*LIST).CPU-QUOTA	I	<integer 1..99>	1,2,9
CPU quota of the VM (exact)	var(*LIST).FINE-CPU-QUOTA	S	<fixed 0.01..99.99>	1,2,9
Effective CPU share of the VM	var(*LIST).EFFECTIVE-CPU-QUOTA	S	<fixed 0.01..100.00>	1,2,9
Maximum CPU utilization of the VM (integer)	var(*LIST).MAX-CPU-UTILIZATION	I	<integer 1..100>	1,2,9
Maximum CPU utilization of the VM (exact)	var(*LIST).FINE-MAX-UTIL	S	<fixed 0.01..100.00>	1,2,9
Maximum CPU utilization of the VM (effective)	var(*LIST).EFFECTIVE-MAX-UTIL	S	<fixed 0.01..100.00>	1,2,9
Name of the VM group	var(*LIST).VM-GROUP	S	*NONE <name 1..8>	1,2,9
Is the IO-RESET privilege assigned?	var(*LIST).IO-PRIVIL	B	TRUE / FALSE	1,5,9
Is the IO-RESET privilege assigned?	var(*LIST).IO-PRIO-PRIVIL	B	TRUE / FALSE	1,5,9
Is the ASSIGN-BY-GUEST privilege assigned for at least one assignment set?	var(*LIST).ASS-BY-GUEST-PRIVIL	B	TRUE / FALSE	1,5,9
Is the AUTO-SNAP-ASSIGNMENT privilege assigned?	var(*LIST).AUTO-SNAP-PRIVIL	B	TRUE / FALSE	1,5,9
Size of VM main memory in Mbytes	var(*LIST).MEM-SIZE	S	<integer 1..1048576>	1,8,9
Minimum size of VM main memory in Mbytes	var(*LIST).MIN-MEM-SIZE	S	<integer 1..1048576> *UNDEF	1,8,9
Maximum size of the VM's main memory in Mbytes	var(*LIST).MAX-MEM-SIZE	S	<integer 1..1048576> *UNDEF	1,8,9
New main memory size for VM for main memory reconfiguration in Mbytes	var(*LIST).PLANNED-MEM-SIZE	S	*UNDEF <integer 1..1048576>	1,8,9
Size of the main memory for BS2000/OSD in Mbytes	var(*LIST).BS2-MEM-SIZE	I	<integer 1..1048576>	1,8,9
Maximum I/O utilization of the VM	var(*LIST).MAX-IO-UTILIZATION	I	<integer 1..100>	1.9

Output information	Name of the S variable	T	Contents	Condition
Is the I/O utilization monitored by IOLVM?	var(*LIST).MAX-IO-UTIL-ACTIVE	B	TRUE / FALSE	1.9
Status of VM	var(*LIST).STA	S	*INIT-ONLY / *RUN *WAIT / *DOWN / *MIGRATING	1.9
Is the VM in the wait state following a global /HOLD-VM from the VM2000 administrator?	var(*LIST).WAIT-GLOBAL	B	TRUE / FALSE	1.9
Is the VM in the wait state following a selective /HOLD-VM from the VM2000 administrator?	var(*LIST).WAIT-SEL	B	TRUE / FALSE	1.9
Is the VM in the wait state following a global /HOLD-VM from the VM administrator?	var(*LIST).WAIT-BY-VM-ADM	B	TRUE / FALSE	1.9
Is the VM in the VM2000-internal wait state?	var(*LIST).WAIT-INTERNAL	B	TRUE / FALSE	1.9
Is a VM administrator connected?	var(*LIST).VM-ADM	B	TRUE / FALSE	1.9
Guest operating system	var(*LIST).GUEST-TYPE	S	BS2-OSD *UNKNOWN	1.9
Version of the guest operating systems	var(*LIST).GUEST-VERSION	S	<c-string 1..8> *UNKNOWN	1.9
Scheduling procedures	var(*LIST).SCHEDULE	S	*NONE *BY-TIMESLICE *DEDICATED-CPUS	2.9
CPU quota (exact) of the VM group when the VM is assigned to a VM group	var(*LIST).VM-GROUP-CPU-QUOTA	S	*NONE <fixed 0.01..99.99>	2.9
Setting for VM-ACTIVE-IDLE	var(*LIST).VM-ACTIVE-IDLE	S	*NO *AT-DEDICATED-CPUS	2.9
Virtual CPUs of the VM	var(*LIST).VIRT-CPU(*LIST).NUM	S	<x-text 2..2>	2.9
Status of virtual CPUs of VM	var(*LIST).VIRT-CPU(*LIST).STA	S	*RUN / *IDLE / *INIT *WAIT / *STOP / *HALT *BLOCK / *SLEEP *OFFLINE	2.9
Partition of global storage assigned to the VM	var(*LIST).PART-ID	S	<name 1..8> / *REAL *VIRTUAL / *NONE	3.9
Usage mode of the global storage of the VM	var(*LIST).GS-ATTR	S	*SHARE / *EXCL *UNDEF	3.9
Size of VM global storage in Mbytes	var(*LIST).GS-SIZE	I	<integer 0..134 217 728>	3.9

Output information	Name of the S variable	T	Contents	Condition
Mode of the global storage of the VM	var(*LIST).GS-MODE	S	*MONO / *DUAL *NONE	3.9
State of the global storage of the VM	var(*LIST).GS-STA	S	*ASSIGN *REMOVE / *NONE	3.9
Monitor job variable for monitoring the VM	var(*LIST).VM-MONJV	S	<name 1..54> / *NONE	4.9
Monitor job variable for monitoring the reconfiguration of the main memory for the VM	var(*LIST).RECONF-MONJV	S	<name 1..54> / *NONE	4.9
Command/function scope for VM2000 administrator (monitor VM) and VM administrator	var(*LIST).ADM-PRIVIL	S	*VM2000-MIN / *VM2000-STD / *VM-STD / *VM-MAX	5.9
Specification of the assignment sets for the VM ASSIGN-BY-GUEST=*YES(. . .)	var(*LIST).ASSIGNMENT-SET-SPEC	S	*NONE / *ALL / *EXCEPT / *LIST	5.9
List of the assignment sets for the VM (only *EXCEPT / *LIST)	var(*LIST).ASSIGNMENT-SET(*LIST)	S	*STDSET / <name 1..8>	5.9
Number of real normal CPUs of the server	var(*LIST).NUM-OF-NORMAL-CPUS	I	<integer 1..64>	6
Number of extra CPUs of the server	var(*LIST).NUM-OF-EXTRA-CPUS	I	<integer 0..63>	6
Number of spare CPUs of the server	var(*LIST).NUM-OF-SPARE-CPUS	I	<integer 0..63>	6
Maximum possible number of virtual CPUs for a VM	var(*LIST).MAX-NUM-CPUS-PER-VM	I	<integer 1..32>	6
Total possible number of virtual CPUs	var(*LIST).NUM-OF-VIRT-CPUS	I	<integer 1..3168>	6
Value of the VM2000 parameter PATH-GROUP-ID	var(*LIST).PATH-GROUP-ID-MODE	S	*COMPLEX *FROM-SYSID	6
Displays whether VM2000 is running on a Server Unit in an HA cluster	var(*LIST).HA-CLUSTER	B	TRUE / FALSE	6
Size of the main memory available for VM2000 in Mbytes	var(*LIST).REAL-MEM-SIZE	I	<integer 1..1048576>	6
Minimum size of main memory required by VM2000 in Mbytes	var(*LIST).REAL-MIN-MEM-SIZE	I	<integer 1..1048576>	6
Size of the global storage available for VM2000 in Mbytes	var(*LIST).GS-SIZE	I	<integer 0..134217728>	6
Main memory size of all XenVMs (including domain 0) in Mbytes	var(*LIST).FOREIGN-MEM-SIZE		<integer 0..1048576>	6
VM2000 hypervisor main memory size in Mbytes	var(*LIST).HYPERVISOR-SIZE	I	<integer 0..255>	6

Output information	Name of the S variable	T	Contents	Condition
Start address of the VM2000 hypervisor main memory in Mbytes	var(*LIST).HYPERVISOR-LOCATION	S	<x-text 1..8> *SPREAD	6
Real CPUs of the server	var(*LIST).REAL-CPU(*LIST).ID	S	<x-text 2..2>	6
CPU pool to which a real CPU belongs	var(*LIST).REAL-CPU(*LIST).CPU-POOL	S	<name 1..8> *STDPOOL / *NONE / *POOL0 / *FOREIGN	6
Status of the real CPUs of the server	var(*LIST).REAL-CPU(*LIST).STA	S	*ATT / *DET / *ERR / *OFFLINE *WAIT-FOR-MAINT	6
Type of the real CPUs of the server	var(*LIST).REAL-CPU(*LIST).ATTR	S	*NORMAL / *EXTRA *FOREIGN / *SPARE *UNUSED	6
Max. number of VMs for this server	var(*LIST).MAX-VMS	I	<integer 1..99>	6
Max. number of VM groups for this server	var(*LIST).MAX-VM-GROUPS	I	<integer 0..50>	6
Max. number of CPU pools for this server	var(*LIST).MAX-CPU-POOLS	I	<integer 1..64>	6
Default value for the assignment set (VM2000 parameter ASSIGN-BY-GUEST=*NO/*YES)	var(*LIST).ASSIGNMENT-SET-DEF	S	*NONE / *STDSET	6
Assignment set for new devices	var(*LIST).ASSIGNMENT-SET-NEW	S	*NONE / *STDSET / <name 1..8>	6
Devices of the VM	var(*LIST).UNIT(*LIST).MN	S	<alphanum-name 2..2> <x-text 4..4>	7.9
Who assigned the device to the VM?	var(*LIST).UNIT(*LIST).ASS-BY	S	*VM-ADMIN *GUEST / *UNDEF	7.9
Usage mode of the device	var(*LIST).UNIT(*LIST).USAGE	S	*EXCL / *SHARED *UNDEF	7.9
Is the device a virtual console?	var(*LIST).UNIT(*LIST).VC	B	TRUE / FALSE	7.9
Is the device in KEEP mode?	var(*LIST).UNIT(*LIST).KEEP	B	TRUE / FALSE	7.9
Start address of VM main memory in Mbytes	var(*LIST).LOCATION	S	<x-text 1..8> *SPREAD	8.9
Devices not assigned to a VM	var(*LIST).UNIT(*LIST).MN	S	<alphanum-name 2..2> <x-text 4..4>	10.12
Usage mode of the free device	var(*LIST).UNIT(*LIST).USAGE	S	*SHARED / *UNDEF	10.12
Is the free device a virtual console?	var(*LIST).UNIT(*LIST).VC	B	TRUE / FALSE	10.12
Is the device in KEEP mode?	var(*LIST).UNIT(*LIST).KEEP	B	TRUE / FALSE	10.12

Output information	Name of the S variable	T	Contents	Condition
Largest possible main memory size for a new VM in Mbytes (see note on page 598)	var(*LIST).LARG-FREE-DOM	I	<integer 0..1048576>	11.12
Start address of a free main memory in Mbytes	var(*LIST).MEM-SLICE(*LIST).BEGIN	S	<x-text 1..8> *SPREAD	11.12
End address of the free main memory in Mbytes	var(*LIST).MEM-SLICE(*LIST).END	S	<x-text 1..8> *SPREAD	11.12
Size of the free main memory in Mbytes	var(*LIST).MEM-SLICE(*LIST).SIZE	I	<integer 1..1048576>	11.12
Name of the Server Unit in the HA cluster	var(*LIST).SERVER_UNIT-NAME	S	<name 1..64>	13
Size of the main memory (in Mbytes) which is occupied for non-highly available XenVMs on the Server Unit in the HA cluster	var(*LIST).FOREIGN-MEM-SIZE-NHA	I	<integer 0..1048576>	13
Size of the main memory (in Mbytes) which is occupied for highly available XenVMs on the Server Unit in the HA cluster	var(*LIST).FOREIGN-MEM-SIZE-HA	I	<integer 0..1048576>	13
Number of BS2000 VMs which can still be configured on the local Server Unit in the HA cluster	var(*LIST).FREE-VMS	I	<integer 0..98>	13
Conflict in the HA cluster because of the number of BS2000 VMs which can still be configured on the local Server Unit in the HA cluster?	var(*LIST).FREE-VMS-CONFL	B	TRUE / FALSE	13
Largest possible main memory size (in Mbytes) with which a VM can still be configured on the local Server Unit in the HA cluster	var(*LIST).FREE-MEM-SIZE	I	<integer 0..1048576>	13
Conflict in the HA cluster because of the largest possible main memory size with which a VM can still be configured on the local Server Unit in the HA cluster?	var(*LIST).FREE-MEM-SIZE-CONFL	B	TRUE / FALSE	13
Conflict in the HA cluster because of the VM's specified index?	var(*LIST).VM(*LIST).SPEC-VM-INDEX-CONFL	B	TRUE / FALSE	13
Specified index for the VM on the Server Unit in the HA cluster (0: *ANY)	var(*LIST).VM(*LIST).SPEC-VM-INDEX	I	<integer 0..99>	13

Output information	Name of the S variable	T	Contents	Condition
Current index of the VM on the Server Unit in the HA cluster (0: VM not started after failover)	var(*LIST).VM(*LIST).CURR-VM-INDEX	I	<integer 0..99>	13
Name of the VM on the Server Unit in the HA cluster	var(*LIST).VM(*LIST).VM-NAME	S	<name 1..8>	13
Is the VM on the Server Unit in the HA cluster highly available?	var(*LIST).VM(*LIST).HA	B	TRUE / FALSE	13
Name of the VM's CPU pool on the Server Unit in the HA cluster	var(*LIST).VM(*LIST).CPU-POOL	S	*STDPOOL / <name 1..8>	13
Multiprocessor level of the VM on the Server Unit in the HA cluster	var(*LIST).VM(*LIST).NUM-OF-CPUS	I	<integer 1..32>	13
Size of the main memory of the VM on the Server Unit in the HA cluster in Mbytes	var(*LIST).VM(*LIST).MEM-SIZE	I	<integer 1..1048576>	13
Output of the assignment sets for the VM on the Server Unit in the HA cluster	var(*LIST).VM(*LIST). ASSIGNMENT-SET-SPEC	S	*NONE / *ALL / *EXCEPT / *LIST	13
List of the assignment sets for the VM on the Server Unit in the HA cluster	var(*LIST).VM(*LIST).ASSIGNMENT- SET(*LIST)	S	*STDSET / <name 1..8> / ' '	13

Output format for INFORMATION = *STD*Example*

VM-ID	CPU-POOL(P)	VM-GROUP(G)	CPU-Q	MAX-U	PROC	MEM-SIZE	MAX-IO
1	MONITOR	*STDPOOL(P)	30.00	100.00	2(+1)	128 MB	(100)
2	SYSTEM2	*STDPOOL(P)	20.00	100.00	2(+1)	128 MB	(100)
3	SYSTEM3	*STDPOOL(P)	50.00	100.00	4(+1)	256 MB	(100)

VM-ID	STATE	GUEST	VERSION	ADMIN	PRIV
1	MONITOR	RUNNING	BS2-OSD V09.0A	YES	AG,AS
2	SYSTEM2	RUNNING	BS2-OSD V08.0A	NO	AS
3	SYSTEM3	INIT-ONLY	UNKNOWN	NO	AS

F R E E VM:12 LARGEST-DOMAIN: 3558 MB

Meaning of the output columns

VM-ID

Identifies the VM by means of VM index and VM name.

CPU-POOL(P)

VM-GROUP(G)

Shows whether the VM is assigned to a VM group. If the VM is not assigned to a VM group, the CPU pool to which the VM is assigned is shown. Whether the VM is assigned to a VM group is indicated by a suffix:

<vm-group-name>(G) The VM is assigned to a VM group; the VM's member CPU quota is displayed under CPU-Q.

<cpu-pool-name>(P) The VM is not assigned to a VM group; the CPU quota of the (individual) VM is displayed under CPU-Q.

CPU-Q

CPU quota of the VM as set by means of command. For VMs which belong to a VM group this is the member CPU quota (MEMBER-CPU-QUOTA operand), otherwise the CPU quota of the VM which does not belong to a VM group (CPU-QUOTA operand).

MAX-U

Maximum CPU utilization of the VM as set by means of command (MAX-CPU-UTILIZATION operand).

PROC

Virtual CPUs of the VM in the form *m* or *m(+n)*.*m* Number of virtual normal CPUs of the VM (multiprocessor level of the VM)*n* Number of virtual spare CPUs of the VM

MEM-SIZE

Current main memory size of the VM in Mbytes.

MAX-IO

Maximum I/O utilization of the VM as set by means of a command (MAX-IO-UTILIZATION operand). If the value is specified in parentheses, this means it is currently not being monitored by the IORM function IOLVM, see [page 360](#).

STATE

State of the VM. States may be combined by means of a plus sign (e.g. WAIT(GLB+VMA)).

INIT-ONLY	The VM is initialized (after /CREATE-VM or /RESET-VM), but no guest system has yet been started
RUNNING	The guest system on the VM has been started and is running
WAIT(...)	The VM is in the wait state; the guest system is at rest (cause)
WAIT(GLB)	Global HOLD (/HOLD-VM *ALL) of the VM2000 administrator
WAIT(INT)	Internal wait state of the VM, e.g. during /MOVE-VM
WAIT(SEL)	Selective HOLD (/HOLD-VM VM-ID=2..n) of the VM2000 administrator
WAIT(VMA)	/HOLD-VM of the VM administrator for his VM
DOWN	The guest system on the VM is terminated (after shutdown or SETS in the guest system)
MIGRATING	A Live Migration (SQ servers) is running for the VM.

GUEST

Operating system in the guest system (e.g. BS2-OSD). UNKNOWN The operating system is not (yet) known (e.g. in the startup phase).

VERSION

Version of the operating system in the guest system in the form Vnn.na. Blanks are output here if the operating system is not (yet) known (UNKNOWN in GUEST column).

ADMIN

Indicates if a VM administrator is connected to the virtual machine (YES/NO).

PRIV

Privileges of the VM (AG=ASSIGN-BY-GUEST / IO-R=IO-RESET / AS=AUTO-SNAP-ASSIGNMENT / IO-P=IO-PRIORITY).

The number of VMs not yet initialized and the largest free main memory size for a new VM are shown in the last line of the output information, starting with FREE.

- S If the Channel Subsystem Monitoring Facility is activated in one of the selected VMs, the VM2000 administrator is informed of this in the case of INFORMATION=*STD/*ALL by message VMS2035 (see [page 392](#)).
- SQ See the information on FREE on [page 598](#).

Output format for INFORMATION = *CPU*Example*

```

VM-ID          CPU-POOL  GR-CPU-Q  CPU-Q/ EFF-Q  MAX-U/ EFF-U  PROC
1 MONITOR      *STDPOOL                30.00/ 30.00 100.00/ 50.00 2(+1)
2 SYSTEM2      *STDPOOL                20.00/ 20.00 100.00/ 50.00 2(+1)
3 SYSTEM3      *STDPOOL                50.00/ 50.00 100.00/100.00 4(+1)

VM-ID          SCHED ACT-I  VIRT CPUS
1 MONITOR      DC      NO      00(RUN),01(RUN),02(OFF)
2 SYSTEM2      DC      NO      00(RUN),01(RUN),02(OFF)
3 SYSTEM3      -       NO      00(WAIT),01(WAIT),02(WAIT),03(WAIT),04(OFF)

```

Meaning of the output columns

VM-ID / PROC

Corresponds to INFORMATION=*STD.

CPU-POOL

Shows the CPU pool to which the VM is assigned.

GR-CPU-Q

Shows whether the VM is assigned to a VM group.**If the VM is assigned to a VM group, the CPU quota of the VM group is displayed here, and the member CPU quota of the VM under CPU-Q.****If the VM is not assigned to a VM group, blanks are displayed here, and the CPU quota of the (individual) VM under CPU-Q.**

CPU-Q/EFF-Q

CPU quota of the VM in the format <cpu-quota> / <effective-cpu-quota>, where:

<cpu-quota>

CPU quota of the VM as set by means of command. For VMs which belong to a VM group this is the member CPU quota (MEMBER-CPU-QUOTA operand), otherwise the CPU quota of the VM which does not belong to a VM group (CPU-QUOTA op).

<effective-cpu-quota>

Effective CPU share of the VM, calculated by VM2000. Values < 0.01 are rounded up to 0.01.

MAX-U/EFF-U

Maximum CPU utilization of the VM in the format <max-cpu-util> / <effective-max-cpu-util>, where:

<max-cpu-util>

Maximum CPU utilization set by means of command (MAX-CPU-UTILIZATION operand) for the VM.

<effective-max-cpu-util>

Effective maximum CPU utilization of the VM, calculated by VM2000.

SCHED

Scheduling procedure:

- No virtual CPU is executable
- DC Fixed CPU assignment (dedicated CPUs)
- TS Time slicing

ACT-I

Setting for controlling the real CPU in the case of fixed CPU assignment:

- NO The real CPU is removed from the VM if the VM's virtual CPU which runs on this is inactive (interruptible wait state, "Idle").
- AT-DC The VM retains control over the real CPU even if the VM's virtual CPU which runs on this is inactive (interruptible wait state, "Idle", S servers).

VIRT CPUS

Virtual normal and spare CPUs of the VM in the format <cpu-id>(<vstate>),....

- <cpu-id> CPU identifier.
- <vstate> State of the virtual CPU, see [page 216](#)
(RUN / IDLE / INIT / WAIT / STOP / HALT / BLOCK / SLEEP / OFF).

When VM-IDENTIFICATION=*VM2000, the output information for the virtual CPUs of the VMs is followed by the output information for all real CPUs of the server, grouped according to the type of CPU and the CPU pool to which the CPU belongs.

Example

CPU-ATTR	CPU-POOL	REAL CPUS
NORMAL	*STDPOOL	00(ATT),01(ATT),02(ATT),03(ATT)
EXTRA	*STDPOOL	04(DET)
SPARE		05(DET)

Meaning of the output columns

CPU-ATTR

Type of CPU (NORMAL / EXTRA / SPARE / FOREIGN / UNUSED), see [page 213](#).

CPU-POOL

Identifies the CPU pool to which the real CPU belongs by means of its name.

- *STDPOOL Standard CPU pool.
- *POOL0 Pool of domain 0 (SQ servers).
- *FOREIGN Pool for the XenVMs (SQ servers).

REAL CPUS

CPU identification and status in the format <cpu-id>(<rstate>),....

- <cpu-id> Identifier of the real CPU.
- <rstate> Status of the real CPU, see [page 214](#)
(ATT / DET / WFM / ERR / OFF).

Output format for INFORMATION = *UNITS*Example*

```

VM-ID          UNITS
1 MONITOR      ASSIGNED BY VM-ADMIN
                EXCL:   C2,C3,V0,V1,2430,2431,2432,2433
                SHARED: 22F4,22F5,22F6,22F7,58F2
                ASSIGNED BY GUEST
                SHARED: E062
2 SYSTEM2      ASSIGNED BY VM-ADMIN
                EXCL:   CH,CI,E05F,E060,E061
                SHARED: E062
3 SYSTEM3      NO CORRESPONDING DEVICE ASSIGNED
DISPOSABLE     UNASS:   CB(VC),CC(VC),CD(VC),CE(VC),CF,CG,CJ,CK,CL, ...
                SHARED: 22F4,22F5,22F6,22F7,58F2,E062

```

Meaning of the output columns

VM-ID

Identifies the VM by means of VM index and VM name.

DISPOSABLE The devices can be assigned to a VM.

UNITS

The output of the devices assigned to a VM takes place VM-specifically in blocks: first the explicitly assigned devices (ASSIGNED BY VM-ADMIN), then the implicitly assigned devices (ASSIGNED BY GUEST). The output is grouped in accordance with the usage mode (EXCL/SHARED/KEEP). Empty blocks or groups are omitted.

The devices assigned to a VM are output in blocks on a VM-specific basis: first the explicitly assigned devices (BY VM-ADMIN), then the implicitly assigned devices (BY GUEST). The output is grouped in accordance with the usage mode (EXCL/SHARED/KEEP). Empty groups are omitted.

The device list is arranged in ascending order of the value range (two- or four-digit). The two-digit device names are output first, followed by the four-digit device names.

S For S servers the suffix (VC) identifies a device that is used or can be used as a virtual console (see [page 117](#)).

If the VM is not assigned to a device which corresponds to the command call, NO CORRESPONDING DEVICE ASSIGNED is output.

If UNITS=list-poss(28): <...> is specified (without the use of wildcards), an appropriate message is output for each device affected.

Output format for INFORMATION = *GS*Example*

VM-ID	PARTITION-ID	ATTR	GS-SIZE	MODE	STATUS
1 VM0001	*REAL	SH	16384 MB	DUAL	ASS
2 VM0002	VIRTGS03		256 MB	DUAL	ASS
3 VM0003	VIRTGS01	SH	512 MB	DUAL	ASS
4 VM0004	NO GS	ASSIGNED			

Meaning of the output columns

VM-ID

Identifies the VM by means of VM index and VM name.

PARTITION-ID

Describes the assigned partition.

*REAL The entire global storage is assigned in the real sense.

*VIRTUAL No global storage is currently assigned to the VM. But a global storage partition was assigned to the VM virtually beforehand.

NO GS AVAILABLE No global storage is available at the server.

NO GS ASSIGNED Global storage is available at the server, but no global storage has been assigned to the VM.

ATTR

Usage mode of the global storage.

SH The global storage can be used by several VMs simultaneously (VMSHARED=*YES)

_ The global storage is used by the VM exclusively (VM-SHARED=*NO)

GS-SIZE

Size of the real global storage assigned or of the partition in Mbytes.

MODE

Mode of the partition (MONO/DUAL, see CREATE-GS-PARTITION).

STATUS

Specifies whether the global storage described is still assigned to the VM (ASSIGNED) or has been removed (REMOVED).

Output format for INFORMATION = *MEMORY (S servers)*Example (VM-ID=*VM2000)*

VM-ID	LOC-BEGIN	MEMORY-SIZE		
HYPERVERSOR	00000000	35 MB		
1 MONITOR	00000023	128 /	128 MB	
2 SYSTEM2	000000A3	128 /	128 MB	
NONE	00000123	3537 MB		
3 SYSTEM3	00000EF4	256 /	256 MB	
LARGEST FREE DOMAIN:		3537		

Meaning of the output columns

VM-ID

Identifies the VM by means of VM index and VM name.

NONE Main memory is not used by a VM (free main memory).

LARGEST FREE DOMAIN

Maximum free main memory for a new VM. Corresponds to the output in the last line in the case of INFORMATION=*STD, see [page 592](#).

LOC-BEGIN

Start address of the main memory area in Mbytes.

MEMORY-SIZE

Size of a free memory area in Mbytes or main memory of the VM in the form

<memorysize>/ <min-memory-size> / (<planned-memory-size>) in Mbytes.

<planned-memory-size> is only output when main memory reconfiguration is in progress.

When VM-IDENTIFICATION=*VM2000, the free memory areas are also output. In this case the output lines are arranged in ascending order of the start addresses of the memory areas. Otherwise they are arranged according to the VM index.

Output format for INFORMATION = *MEMORY (SQ servers)*Example (VM-ID=*VM2000)*

VM-ID	BS2-MEM-S	MEM-SIZE	MIN-MEM-S	MAX-MEM-S	DEST-MEM-S
1 MONITOR	3996	4096	2048	8192	
2 SYSTEM2	970	1024	1024	2048	
FOREIGN VMS		9215			
FREE		15625			

Meaning of the output columns

VM-ID

Identifies the VM by means of VM index and VM name.

FOREIGN VMS Main memory size of the XenVMs (including domain 0).

FREE Maximum free main memory for a new VM. Corresponds to the output in the last line in the case of INFORMATION=*STD, see [page 592](#).

When initializing a VM a temporary main memory area is required. This main memory area cannot be used as main memory for the VM; it is released later. Consequently not all the main memory specified with FREE can be used for the VM.

BS2-MEM-S

Size of the main memory (without the VM's firmware component) which is available for BS2000/OSD in Mbytes.

MEM-SIZE

Size of VM main memory in Mbytes.

MIN-MEM-S

Minimum size of the VM's main memory in Mbytes (for main memory reconfiguration).

MAX-MEM-S

Maximum size of the VM's main memory in Mbytes (for main memory reconfiguration).

DEST-MEM-S

Planned size of the VM's main memory in Mbytes when main memory reconfiguration is in progress.

When VM-IDENTIFICATION=*VM2000, the output lines are sorted according to the VM index in ascending order. The memory area of domain 0 and of the XenVMs is subsequently output, as is the largest free memory area for a new VM.

Output format for INFORMATION = *MONJV*Example*

```
VM-ID          VM-MONJV
 2 SYSTEM2     :4IVR:$TSOS.JV-SYS2
 3 VM0003      :4IVR:$TSOS.JV-SYS3

VM-ID          RECONF-MONJV
 2 SYSTEM2     :4IVZ:$TSOS.JV-S2REC
```

Output lines are shown only for VMs which are monitored using a MONJV or which are explicitly specified in the command.

Meaning of the output columns

VM-ID

Identifies the VM by means of VM index and VM name.

VM-MONJV

File name of the MONJV for the VM.

NO VM-MONJV ASSIGNED No MONJV is defined for the MONJV.

RECONF-MONJV

File name of the MONJV for the main memory reconfiguration. If no reconfiguration of the main memory is in progress or no MONJV has been defined for this, the second output block is not output.

Output format for INFORMATION = *PRIVILEGES*Example*

VM-ID	PRIV	DETAILS
1 MONITOR	AG	SET2
	AS	
2 SYSTEM2	AG	*STDSET
	AS	
3 VM0003	AS	

Meaning of the output columns

VM-ID

Identifies the VM by means of VM index and VM name.

PRIV

Privileges of the VM.

NO PRIVILEGE No privileges.

ADM Command/function scope for VM2000 administrator (monitor VM) or VM administrator, see DETAILS.

AG The VM has the ASSIGN-BY-GUEST privilege. The list of the assignment sets assigned is output in DETAILS.

AS AUTO-SNAP-ASSIGNMENT=*YES (default).

IO-P IO-PRIORITY=*YES.

IO-R IO-RESET=*YES.

DETAILS

Standard command/function scope

For PRIV=ADM

*VM2000-MIN ADMIN-PRIVILEGES=*MIN for the monitor VM

*VM-MAX ADMIN-PRIVILEGES=*MAX for a VM other than the monitor VM

Standard command/function scope

For PRIV=AG

*ALL ASSIGN-BY-GUEST=*YES(SET-NAME=*ALL)

*EXCEPT: set1, set2, ...

ASSIGN-BY-GUEST=*YES(SET-NAME=*EXCEPT(...))

set1, set2, ...

ASSIGN-BY-GUEST=*YES(SET-NAME=<list>)

Output format for INFORMATION = *CONFIGURATION*Example (S servers)*

```

MAX NUMBER OF VMS                15
MAX NUMBER OF VM-GROUPS          8
MAX NUMBER OF CPU-POOLS          5
MAX NUMBER OF CPUS PER VM        4
TOTAL NUMBER OF VIRTUAL CPUS     75
TOTAL REAL MEMORY SIZE           4084 MB
TOTAL MIN MEMORY SIZE            4084 MB
TOTAL GS SIZE                    3072 MB
PATH-GROUP-ID MODE               COMPLEX
ASSIGNMENT SET DEFAULT VALUE:    *NONE
                                NEW DEVICES : *NONE

CPU-ATTR    CPU-POOL    REAL CPUS
NORMAL      *STDPOOL    00(ATT),01(ATT),02(ATT),03(ATT)
EXTRA       *STDPOOL    04(DET)
SPARE       *STDPOOL    05(DET)

```

Meaning of the output lines

The following configuration data of the server is displayed in this output block:

- the maximum number of VMs that can be created
- the maximum possible number of VM groups that can be created
- the maximum possible number of CPU pools for VM2000
- the maximum number of virtual CPUs for a VM (multiprocessor level) which may be selected explicitly for a VM
- the size of the main memory in Mbytes
- the minimum main memory size for VM2000 operation in Mbytes.
- the size of the global storage in Mbytes (only if global storage is available). If the global storage is disconnected, "0" is specified as the size.
- the "Path Group Id" (COMPLEX / FROM-SYSID)
- Default value for the implicit device assignment (*NONE / *STDSET). The default value corresponds to the VM2000 parameter `ASSIGN-BY-GUEST=*NO / *YES` (S servers). On SQ servers the default value is always *NONE.

- Default value for the implicit device assignment for new devices (dynamic configuration modification) (*NONE / *STDSET / <name 1..8>). The default value corresponds to the VM2000 parameter ASSIGN-BY-GUEST=*NO / *YES (S servers), provided this has not been modified by means of /MODIFY-VM-DEVICE-ATTRIBUTES UNITS=*NEW-DEVICES.
- information on the real CPUs. For a description of this output format see [page 594](#).

SQ When VM2000 runs on a Server Unit in an HA cluster, this is displayed in an additional output line:

```
...  
PATH-GROUP-ID MODE                COMPLEX  
HA-CLUSTER                      YES  
ASSIGNMENT SET   DEFAULT VALUE: *NONE  
                 NEW DEVICES  : *NONE  
...
```

Output format for INFORMATION = *HA-CLUSTER (SQ servers)*Example*

```

SHIRKHAN (LOCAL SERVER UNIT)
VM-ID          HA  CPU-POOL      PROC  MEM-SIZE      ASSIGNMENT-SETS
BS2000 VMS
  1/ 1 MONITOR   N  *STDPPOOL     2    2048 MB
  -/ 2 PRODUCT1  Y  USERPOOL     4    4096 MB      *STDSET
(!)3/ 3 GUESTVM3 Y  *STDPPOOL     2    1024 MB      *STDSET
  -/ 4 TESTSHIRK N  *STDPPOOL     1    1024 MB      *STDSET, USERSET
FOREIGN VMS
  N  *FOREIGN
  Y  *FOREIGN
      8195 MB
      512 MB
F R E E VM: 11(!)          MEMORY:14180 MB(!)

BAGHEERA (REMOTE SERVER UNIT)
VM-ID          HA  CPU-POOL      PROC  MEM-SIZE      ASSIGNMENT-SETS
BS2000 VMS
  1/ 1 MONITOR   N  *STDPPOOL     2    2048 MB
  -/ 2 PRODUCT2  Y  USERPOOL     4    6144 MB      *EXCEPT:USERSET
  -/ 3 TESTBAGH  N  *STDPPOOL     1     512 MB      USERSET
  6/ 6 GUESTVM6  Y  *STDPPOOL     4    8192 MB      *STDSET
FOREIGN VMS
  N  *FOREIGN
  Y  *FOREIGN
      8192 MB
      1792 MB

```

The output consists of one information block for each Server Unit in the HA cluster. Each information block is introduced by an information line containing the name of the Server Unit and its type:

LOCAL SERVER UNIT Local Server Unit in the HA cluster (first information block).
 REMOTE SERVER UNIT Remote Server Unit in the HA cluster.

In each information block, the header line for the output columns (see next page) is followed by the information on the BS2000 VMs, introduced by the output line `BS2000 VMS`. This is followed by information on the XenVMs, introduced by the output line `FOREIGN VMS`.

The last line of the output information on the local Server Unit in the HA cluster, introduced by `FREE`, displays the number of VMs which have not yet been initialized and the maximum possible main memory size for a new VM on the local Server Unit in the HA cluster.



The suffixed output `(!)` indicates a possible conflict in the HA cluster.

In the case of `VM: (!)` means that the remaining number of VMs on the local Server Unit is lower than the number of highly available BS2000 VMs on the remote Server Unit in the HA cluster.

In the case of `MEMORY: (!)` means that the free main memory on the local Server Unit is less than the total of the main memory sizes of all highly available BS2000 VMs and XenVMs on the remote Server Unit in the HA cluster.

Meaning of the output columns

VM-ID

Identifies the VM by means of VM index and VM name.

The VM index is displayed in the following form:

vorgesehener VM-Index / momentaner VM-Index

Output of "-" in the envisaged VM index means that no fixed VM index was specified (VM-INDEX=*ANY) when the VM was initialized. This also means that the VM can contain a different VM after a failover in the HA cluster or after a Live Migration on the other Server Unit.

Output of "-" in the current VM index means that the highly available VM is currently not running on this Server Unit (there is only a VM configuration file).



The prefixed output (!) indicates a possible conflict in the HA cluster.

The VM index envisaged for a highly available VM is not available on the other Server Unit.

HA

Specifies whether the BS2000 VM or the group of XenVMs has the attribute "high-availability VM" (Y) or not (N).

CPU-POOL

Identifies the CPU pool to which the VM belongs by means of its name.

*STDPOOL Standard CPU pool.

*FOREIGN Pool for the XenVMs.

PROC

Number of virtual CPUs of the VM (multiprocessor level of the VM)

MEM-SIZE

Current main memory size of the VM in Mbytes.

The total of the main memory sizes of for the group concerned is output for XenVMs.



Unlike with MEM-SIZE in the FOREIGN VMS line of the INFORMATION=*MEMORY output (see [page 598](#)), the main memory size of domain 0 is contained in none of the totals.

ASSIGNMENT-SETS

List of the assigned assignment sets.

Example (S servers)

```

/show-vm-resources information=*all _____ (1)
      CPU-POOL(P)
VM-ID   VM-GROUP(G) CPU-Q   MAX-U   PROC   MEM-SIZE  MAX-IO _____ (2)
 1 MONITOR *STDPOOL(P) 30.00 100.00 2(+1) 128 MB (100)
 2 SYSTEM2 *STDPOOL(P) 20.00 100.00 2(+1) 128 MB (100)
 3 SYSTEM3 *STDPOOL(P) 50.00 100.00 4(+1) 256 MB (100)

VM-ID   STATE           GUEST   VERSION ADMIN PRIV _____ (3)
 1 MONITOR RUNNING          BS2-OSD V09.0A YES  AG,AS
 2 SYSTEM2 RUNNING          BS2-OSD V09.0A NO   AS
 3 SYSTEM3 INIT-ONLY       UNKNOWN NO    AS
F R E E VM:12      LARGEST-DOMAIN: 3558 MB

VM-ID   CPU-POOL  GR-CPU-Q CPU-Q/ EFF-Q MAX-U/ EFF-U PROC _____ (4)
 1 MONITOR *STDPOOL          30.00/ 30.00 100.00/ 50.00 2(+1)
 2 SYSTEM2 *STDPOOL          20.00/ 20.00 100.00/ 50.00 2(+1)
 3 SYSTEM3 *STDPOOL          50.00/ 50.00 100.00/100.00 4(+1)

VM-ID   SCHED ACT-I  VIRT CPUS _____ (5)
 1 MONITOR DC    NO    00(RUN),01(RUN),02(OFF)
 2 SYSTEM2 DC    NO    00(RUN),01(RUN),02(OFF)
 3 SYSTEM3 -    NO    00(WAIT),01(WAIT),02(WAIT),03(WAIT),04(OFF)

CPU-ATTR CPU-POOL  REAL CPUS _____ (6)
NORMAL   *STDPOOL  00(ATT),01(ATT),02(ATT),03(ATT)
EXTRA    *STDPOOL  04(DET)
SPARE    *STDPOOL  05(DET)

VM-ID   VM-MONJV _____ (7)
 3 SYSTEM3 :WUT1:$TSOS.MONJV-SYSTEM3

VM-ID   UNITS _____ (8)
 1 MONITOR ASSIGNED BY VM-ADMIN
          EXCL: C2,C3,V0,V1,2430,2431,2432,2433
          SHARED: 22F4,22F5,22F6,22F7,58F2
          ASSIGNED BY GUEST
          SHARED: E062
 2 SYSTEM2 ASSIGNED BY VM-ADMIN
          EXCL: CH,CI,E05F,E060,E061
          SHARED: E062
 3 SYSTEM3 NO CORRESPONDING DEVICE ASSIGNED
DISPOSABLE UNASS: CB(VC),CC(VC),CD(VC),CE(VC),CF,CG,CJ,CK,CL, _____ (9)
          CM,CN,CO,CP,CQ,CR(VC),CS,CT,CU(VC),CV(VC),CW, _____ (10)
          ... (Ausgabe gekürzt) _____ (11)
          SHARED: 22F4,22F5,22F6,22F7,58F2,E062

```

VM-ID	PARTITION-ID	ATTR	GS-SIZE	MODE	STATUS	_____ (12)
1	MONITOR	*REAL	SH	3072 MB	DUAL	ASS

VM-ID	LOC-BEGIN	MEMORY-SIZE	_____ (13)
HYPERVERSOR	00000000	35 MB	
1 MONITOR	00000023	128 /	128 MB
2 SYSTEM2	000000A3	128 /	128 MB
NONE	00000123	3537 MB	
3 SYSTEM3	00000EF4	256 /	256 MB

VM-ID	PRIV	DETAILS	_____ (14)
1 MONITOR	AG	*STDSET	
	AS		
2 SYSTEM2	AS		
3 SYSTEM3	AS		

- (1) The VM2000 administrator queries the status of all VM2000 resources.
- (2) List of attributes of all VMs. Each VM is also assigned a virtual spare CPU.
- (3) List of statuses for the VMs set up. Up to 12 further VMs can be set up. The largest free main memory size for a new VM is 3558 Mbytes.
- (4) Overview of the assignment of VMs to CPU pools and VM groups, and of the shares of server CPU capacity specified by means of command and calculated by VM2000.
- (5) List of virtual normal and spare CPUs.
- (6) List of real normal, extra and spare CPUs.
- (7) VM3 is monitored with a MONJV.
- (8) List of devices assigned.
- (9) List of free devices, arranged by usage mode.
- (10) Possible virtual consoles are identified by the suffix (VC).
- (11) In the device lists the 2-digit mnemonic device names come first, followed by the 4-digit names.
- (12) The global storage is assigned to the monitor VM.
- (13) Distribution of the VM's main memories in VM2000 main memory.
- (14) Privileges of the VMs.

SHOW-VM-STATUS

Output VM2000 monitored data

`/SHOW-VM-STATUS` is used to output the current planned values and the current monitored data for active objects once only or periodically in ongoing VM2000 operation.

This command is reserved for the VM2000 administrator.

Active objects comprise attached normal and extra CPUs, active VMs, CPU pools with attached real CPUs, and VM groups with active VMs (see [page 248](#)).

The monitored data is output once only or periodically:

- The one-off output of monitored data (`PERIOD=*RECENT(...)`) takes place synchronously to command execution and shows the monitored data from the immediate past. For this purpose the VM2000 monitor collects the current monitored data periodically (every 20 seconds). This data is kept for up to one hour. The one-off output can be called by multiple VM2000 administrators simultaneously, also in parallel to the periodic output of monitored data. Monitored data is output only for VMs, VM groups and CPU pools which existed at the time the command was entered.
- The periodic output of monitored data (`PERIOD=<integer>`) takes place asynchronously wherever the corresponding `/SHOW-VM-STATUS` command was entered. The periodic output cannot be used by more than one VM2000 administrator simultaneously.

The periodic output is terminated by entering a new `/SHOW-VM-STATUS PERIOD=*NO` command from the same input medium (ADMIN dialog of the monitor VM or BS2000 console) or by terminating the associated ADMIN dialog or BS2000 console.

The periodic output is also aborted by deleting a CPU pool or a VM group or by terminating a VM if the command was entered explicitly for this CPU pool (by means of `VM-ID` or `CPU-ID`), this VM group or this VM.

When one VM or multiple VMs are selected (by specifying wildcards), the periodic output is terminated if the specified VM is terminated or if no VM from the selected set exists.

The command outputs information in relation to the attached real CPUs and the active VMs and the created VM groups and CPU pools:

- Current planned values for distributing the CPU capacity of and the CPU time used by the virtual machines. Effective planned values under ideal conditions for active and inactive objects are displayed using `/SHOW-VM-ATTRIBUTES/-RESOURCES` (see [page 239](#)).

On S servers the following information is also supplied:

- CPU time used and the IDLE time of the VM2000 hypervisor
- number of events handled via the VM2000 hypervisor
- values for scheduling by VM2000

SHOW-VM-STATUS

```

INFORMATION = *STD / *TIMER / *COUNTER / *SCHEDULE / *ALL
,PERIOD = *NO / *RECENT(...) / <integer 5..3600> seconds
    *RECENT(...)
        | SECONDS = <integer 1..3600> seconds
,VM-IDENTIFICATION = *ALL / <integer 1..99> / <name 1..8 with-wild(20)>
                    *NONE / *BY-VM-GROUP(...) / *BY-CPU-POOL(...)
    *BY-VM-GROUP(...)
        | VM-GROUP-NAME = <name 1..8>
    *BY-CPU-POOL(...)
        | CPU-POOL-NAME = *STDPOOL / <name 1..8>
,CPU-IDENTIFICATION = *STD / *NONE / *ALL / list-poss(64):<x-text 1..2> / *BY-CPU-POOL(...)
    *BY-CPU-POOL(...)
        | CPU-POOL-NAME = *BY-VM-IDENTIFICATION / *STDPOOL / <name 1..8>

```


INFORMATION =

Specifies the output information.

INFORMATION = *STD

The default value is used (*TIMER).

INFORMATION = *TIMER

The CPU time utilized or the IDLE time and the current planned values are output.

INFORMATION = *COUNTER

The number of events is output (S servers).

INFORMATION = *SCHEDULE

The values for scheduling are output (S servers).

INFORMATION = *ALL

All information for the outputs *TIMER, *COUNTER and *SCHEDULE.

PERIOD =

Defines the intervals at which output is to occur.

PERIOD = *NO

Terminates periodical collection of monitored data and the current output of /SHOW-VM-STATUS PERIOD=<seconds>. Other operands are not relevant in this case.

PERIOD = *RECENT(...)

Triggers one-off output of the monitored data from the immediate past.

SECONDS = <integer 1..3600> seconds

Period (in seconds) for which the monitored data is output.



If VM2000 operation has not run this long, the monitored data from the shorter period is output.

PERIOD = <integer 5..3600> seconds

Starts periodic collection of monitored data and defines the period (in seconds) during which the monitored data is gathered and then output.

VM-IDENTIFICATION =

Identification of the virtual machine.

VM-IDENTIFICATION = *ALL

Information on all active virtual machines is output.

VM-IDENTIFICATION = *NONE

No information on the active virtual machines is output (S servers).

VM-IDENTIFICATION = <integer 1..99> / <name 1..8 with-wild(20)>

Identifies the virtual machine by means of a VM index or VM name.

VM-IDENTIFICATION = *BY-VM-GROUP(...)

Information is output for all active VMs which belong to the VM group (S servers).

VM-GROUP-NAME = <name 1..8>

Name of the VM group.

VM-IDENTIFICATION = *BY-CPU-POOL(...)

Information is output for all active VMs which belong to the CPU pool.

CPU-POOL-NAME = *STDPOOL / <name 1..8>

Name of the CPU pool.

CPU-IDENTIFICATION =

Selects the real CPUs.

CPU-IDENTIFICATION = *STD

[S] When VM-IDENTIFICATION=*ALL/*NONE, information is output on all attached real CPUs. Otherwise no information on real CPUs is output (operand value provided for reasons of compatibility with earlier VM2000 versions).

[SQ] No information on real CPUs is output.

CPU-IDENTIFICATION = *ALL

Information on all attached real CPUs is output.

CPU-IDENTIFICATION = *NONE

No information on real CPUs is output.

CPU-IDENTIFICATION = list-poss(64):<x-text 1..2>

Information on the specified real CPUs is output (S servers).

CPU-IDENTIFICATION = *BY-CPU-POOL(...)

Information on the attached real CPUs in a CPU pool is output (S servers).

CPU-POOL-NAME =

Selects the CPU pool.

CPU-POOL-NAME = *BY-VM-IDENTIFICATION

The CPU pools to which the VMs are assigned (VM-IDENTIFICATION operand) are selected.

CPU-POOL-NAME = *STDPOOL / <name 1..8>

Name of the CPU pool.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
1	0	CMD0001	No action required (not when PERIOD=*RECENT)
2	0	VMS4403	No VM belongs to VM group (not when PERIOD=*RECENT)
2	0	VMS4603	No CPU has been assigned to CPU pool (not when PERIOD=*RECENT)
2	0	VMS4613	No VM has been assigned to CPU pool (not when PERIOD=*RECENT)
	64	VMS3002	Invalid operand value (CPU-IDENTIFICATION)
	64	VMS3015	Monitoring data collection already started
	64	VMS3018	No monitoring object specified
	64	VMS3019	No VM with appropriate name found
	64	VMS4000	VM not yet created
	64	VMS4043	CPU not known in system
	64	VMS4400	VM group not yet created
	64	VMS4600	CPU pool not yet created

Output format for INFORMATION = *STD / *TIMER

The output consists of two information blocks:

- CPU TIMER INFORMATION FOR REAL CPU(S) (S servers)

The CPU-specific information block provides information on time values (actual values for CPU time utilization) of the CPU pools and their attached real CPUs. The information lines are grouped according to CPU pools, and in these according to the CPU-IDs.
- CPU TIMER INFORMATION FOR VM(S)

The VM-specific information block provides information on current planned values and time values (actual values for CPU time utilization) of the CPU pools, the VM groups (S servers), the running VMs and their active virtual CPUs (S servers). The information lines are grouped according to CPU pools, and in these according to VMs which do not belong to a VM group, and VM groups. The virtual CPUs of a VM are arranged according to the CPU-IDs.

Any changes during the monitoring period are marked on the left of the corresponding value lines with * or # to indicate monitored data that has possibly been corrupted.

- * indicates changes which have resulted in falsified monitored data for ACTIVE or IDLE times. These were triggered by:
 - attaching or detaching a real CPU in a CPU pool
 - assigning or removing an (individual) VM or VM group to or from a CPU pool
 - assigning or removing a real (individual) VM to or from a CPU pool

and on S servers by:

 - change of status of a virtual CPU from or to RUNNING status
 - assigning or removing a VM to or from a VM group
 - assigning or removing a VM group to or from a CPU pool
- # indicates explicit or implicit changes to the current CPU share (CUR-Q) or to the maximum CPU utilization (CUR-U). ACTIVE times then no longer match the planned values.

When * and # apply for a values line, * is output.

If the command was specified explicitly for a CPU pool, a VM group, a VM or a real CPU but no real or virtual CPUs are active or belong to the pool, a corresponding message (NO CPU ACTIVE) is output in the information line.

Example of the CPU-specific information block (S servers)

CPU TIMER INFORMATION FOR REAL CPU(S)

CPU-POOL	REAL	HPV-ACTIVE		HPV-IDLE		ALL-VMS-ACTIVE	
	CPU	TIME/	%	TIME/	%	TIME/	%
*STDPPOOL		0.091/	0.07	124.929/	99.66	0.338/	0.27
	00	0.087/	0.14	62.266/	99.34	0.326/	0.52
	01	0.004/	0.01	62.663/	99.97	0.012/	0.02
EXCLPOOL		0.066/	0.05	0.000/	0.00	125.292/	99.95
	02	0.033/	0.05	0.000/	0.00	62.646/	99.95
	03	0.033/	0.05	0.000/	0.00	62.646/	99.95

Meaning of the output columns

CPU-POOL

Name of the CPU pool.

*STDPPOOL Outputs for the standard CPU pool.

REAL CPU

Real CPU identifier (CPU-ID).

HPV-ACTIVE TIME / %

CPU time utilized by the VM2000 hypervisor for the CPU pool or for the real CPU in the form of an absolute value (in seconds) and a percentage value of the CPU capacity of the CPU pool or of the real CPU.

HPV-IDLE TIME / %

IDLE time of the VM2000 hypervisor for the CPU pool or for the real CPU in the form of an absolute value (in seconds) and a percentage value of the CPU capacity of the CPU pool or of the real CPU.

ALL-VMS-ACTIVE TIME / %

CPU time utilized by all VMs for the CPU pool or for the real CPU in the form of an absolute value (in seconds) and a percentage value of the CPU capacity of the CPU pool or of the real CPU.

Example of the VM-specific information block (S servers)

CPU TIMER INFORMATION FOR VM(S)

CPU-POOL	VM-GROUP	VM-ID	VIRT CPU	CPU-Q	CUR-Q	CUR-U	VM - ACTIVE TIME / %
*STDPOOL(P)					50.00	50.00	0.331/ 0.13
1 MONITOR				20.00	25.00	25.00	0.319/ 0.13
			00				0.319/ 0.12
EXTGROUP(G)				80.00	25.00	25.00	0.012/ 0.00
2 SYSTEM2				70.00	25.00	25.00	0.012/ 0.00
			00				0.012/ 0.00
EXCLPOOL(P)					50.00	50.00	125.351/49.99
4 SYSTEM4				1.00	50.00	50.00	125.351/50.00
			00				62.676/24.99
			01				62.675/24.99

Meaning of the output columns

CPU-POOL

Identifies a CPU pool by its name (suffix: (P)).

VM-GROUP

VM-ID

Identifies a VM group (suffix: (G)) by its name or a VM by the VM index and VM name.

VIRT CPU

Identifier of the VM's virtual CPU (CPU-ID).

CPU-Q

CPU quota of the VM or VM group or member CPU quota of the VM in the VM group, set by means of command.

CUR-Q

Calculated current CPU share of the VM, VM group or CPU pool (current anticipated CPU utilization if all VMs were CPU-intensive).

For a CPU pool: Identical to CUR-U.

For a VM group: Relative to the server's current CPU capacity.

For a VM: Relative to the server's current CPU capacity. Identical to the planned value in the VM2000 report of openSM2.

CUR-U

Calculated current maximum CPU utilization of the VM, VM group or CPU pool.

For a CPU pool according to the formula: Number of attached CPUs in the CPU pool / number of attached server CPUs (in percent).

For a VM group: Current maximum CPU utilization of the VM group.

For a VM: Current maximum CPU utilization of the VM.

Identical to the UTIL value in the VM2000 report of openSM2.

VM-ACTIVE TIME / %

CPU time utilization in the form of an absolute value (in seconds) and as a percentage value (actual values).

For a CPU pool:

Total CPU times of all VMs in the CPU pool. Percentage value relative to the current server CPU capacity.

For a VM group:

Total CPU times of all VMs in the VM group. Percentage value relative to the current server CPU capacity.

For a VM:

Total CPU times of all the VM's active virtual CPUs. Percentage value relative to the current server CPU capacity.

For a virtual CPU:

CPU time of the virtual CPU. Percentage value relative to the current server CPU capacity.

Example of the VM-specific information block (SQ servers)

```

CPU TIMER INFORMATION FOR VM(S)
CPU-POOL  NUM OF  CPU-Q   CUR-Q   CUR-U   VM - ACTIVE
VM-ID      CPUS
*POOL0     4           34.087/ 14.18
*STDPPOOL  3           100.00  100.00  30.078/ 16.68
  1 MONITOR 2    14.00   66.66   66.66   1.101/  0.61
  2 SYSTEM2 1    50.00   33.33   33.33   28.097/ 15.58

```

Meaning of the output columns

CPU-POOL

Identifies a CPU pool by means of its name. The pool of domain 0 is named *POOL0.

VM-ID

Identifies a VM by means of VM index and VM name.

NUM OF CPUS

Number of real CPUs currently available in this CPU pool or number of active virtual CPUs of the VM.

CPU-Q

CPU quota of the VM, set by means of a command.

CUR-Q

Calculated current CPU share of the VM or of the CPU pool (current expected CPU utilization if all VMs were CPU-intensive).

For a CPU pool (with the exception of *POOL0): identical to CUR-U.

For a VM: Relative to the current CPU capacity of the CPU pool. Identical to the planned value in the VM2000 report of openSM2.

CUR-U

Calculated current maximum CPU utilization of the VM or of the CPU pool.

For a CPU pool (with the exception of *POOL0), according to the formula: Number of attached CPUs in the CPU pool / Number of attached CPUs of the server (in percent).

For a VM: Current maximum CPU utilization of the VM.

Identical to the UTIL value in the VM2000 report of openSM2.

VM-ACTIVE TIME / %

The output consists of two information blocks:

For a CPU pool:

Total CPU times of all VMs in the CPU pool. Percentage value relative to the current server CPU capacity.

For a VM:

CPU time of the VM (measured by Xen). Percentage value relative to the current server CPU capacity.

Output format for INFORMATION = *COUNTER (S servers)

The output consists of two information blocks:

- **CPU COUNTER INFORMATION FOR REAL CPU(S)**
The CPU-specific information block provides information on counter values of the attached real CPUs.
The information lines are grouped according to CPU pools, and in these according to the CPU-IDs.
- **CPU COUNTER INFORMATION FOR VM(S)**
The VM-specific information block provides information on counter values of the active virtual CPUs of the running VMs.
The information lines are grouped according to CPU pools, and in these according to VMs. The virtual CPUs of a VM are arranged according to the CPU-IDs.

Any changes during the monitoring period are marked in the appropriate place with * to indicate monitored data that has possibly been corrupted, see [page 612](#).

Example of the CPU-specific information block

```

CPU COUNTER INFORMATION FOR REAL CPU(S)
CPU-POOL      REAL    I-O INT.  INTERCEPT.  EMULATION      SCHEDULE
              CPU     COUNT/ %   COUNT/ %      COUNT/ %      COUNT/ %
*STDPOOL      00     429/100   1325/ 50      4/100         1325/ 50
              01         0/ 0      63/ 2         0/ 0          63/ 2
EXCLPOOL      02         0/ 0      633/ 24       0/ 0          633/ 24
              03         0/ 0      633/ 24       0/ 0          633/ 24

```

Meaning of the output columns

Percentage values in the CPU-specific information block relate to the total of the values for all attached real CPUs of the server.

CPU-POOL

Name of the CPU pool to which the real CPU belongs.

*STDPOOL The CPU belongs to the standard CPU pool.

REAL CPU

Real CPU identifier (CPU-ID).

I-O INT. COUNT / %

Number/percentage of I/O interrupts handled for a VM via the VM2000 hypervisor

INTERCEPT. COUNT / %

Number/percentage of interrupt events which were not caused by a command emulation and which lead to a switch from VM mode to VM2000 hypervisor mode.

EMULATION COUNT / %

Number/percentage of command emulations handled for a VM via the VM2000 hypervisor.

SCHEDULE COUNT / %

Number/percentage of time slice allocations to a VM.

Example of the VM-specific information block

CPU COUNTER INFORMATION FOR VM(S)

CPU-POOL	VIRT	I-O	INT.	INTERCEPT.	EMULATION	SCHEDULE	
VM-ID	CPU	COUNT/	%	COUNT/	%	COUNT/	%
*STDPPOOL							
1	MONITOR	00	429/100	1325/ 50	4/100	1325/ 50	
2	SYSTEM2	00	0/ 0	63/ 2	0/ 0	63/ 2	
EXCLPOOL							
4	SYSTEM4	00	0/ 0	633/ 24	0/ 0	633/ 24	
		01	0/ 0	633/ 24	0/ 0	633/ 24	

Meaning of the output columns

Percentage values in the VM-specific information block relate to the total of the values for all active virtual CPUs of all VMs.

CPU-POOL

VM-ID

Identifies a CPU pool by means of its name or a VM by means of the VM index and VM name.

VIRT CPU

Virtual CPU identifier (CPU-ID).

NO CPU ACTIVE No virtual CPU of the VM has run in this time period.

I-O INT. COUNT / %

Number/percentage of I/O operations handled for the virtual CPU via the VM2000 hypervisor.

INTERCEPT. COUNT / %

Number/percentage of interrupt events which were not caused by a command emulation and which lead to a switch from VM mode to VM2000 hypervisor mode for the virtual CPU.

EMULATION COUNT / %

Number/percentage of command emulations handled for the virtual CPU via the VM2000 hypervisor.

SCHEDULE COUNT / %

Number/percentage of time slice allocations to the virtual CPU.

Output format for INFORMATION = *SCHEDULE (S servers)

The output consists of two information blocks:

- **CPU SCHEDULE INFORMATION FOR REAL CPU(S)**
The CPU-specific information block provides information on scheduling values of the attached real CPUs.
The information lines are grouped according to CPU pools, and in these according to the CPU-IDs.
- **CPU SCHEDULE INFORMATION FOR VM(S)**
The VM-specific information block provides information on scheduling values of the active virtual CPUs of the running VMs.
The information lines are grouped according to CPU pools, and in these according to VMs. The virtual CPUs of a VM are arranged according to the CPU-IDs.

Any changes during the monitoring period are marked in the appropriate place with * to indicate monitored data that has possibly been corrupted, see [page 612](#).

Example of the CPU-specific information block

```
SCHEDULE INFORMATION FOR REAL CPU(S)
CPU-POOL   REAL   <-- TIME SLICE  -->  NON-AFFIN SCHED
           CPU    AVG %RUNOUT %SHORT  %IDLE %LOAD
*STDPOOL   00    0.24      1      62  DEDICATED SCHED
           01    0.19      0       0  DEDICATED SCHED
EXCLPOOL   02    VM-ACTIVE-IDLE  DEDICATED SCHED
           03    VM-ACTIVE-IDLE  DEDICATED SCHED
```

Meaning of the output columns

Percentage values in the CPU-specific information block relate to the total of the values for all scheduling procedures of the real CPUs.

CPU-POOL

Name of the CPU pool to which the real CPU belongs.

*STDPOOL The CPU belongs to the standard CPU pool.

REAL CPU

Real CPU identifier (CPU-ID).

TIME SLICE AVG

Average value of the time slices (in milliseconds) of the virtual CPUs on the real CPU.

TIME SLICE %RUNOUT

Percentage value of the completely used time slices of the virtual CPUs.

TIME SLICE %SHORT

Percentage value of the little-used time slices of the virtual CPUs (value less than 0.1 milliseconds or less than a tenth of the time slice).



In the event of fixed CPU assignment (scheduling procedure “dedicated CPUs”), the value `VM-ACTIVE-IDLE` is output in the `<-- TIME SLICE -->` column if this parameter is effective for the VM.

NON-AFFIN SCHED %IDLE

Percentage value of the non-CPU-affined scheduling operations on the real CPU for avoiding an IDLE status of the real CPU despite executable (non-affined) virtual CPUs.

NON-AFFIN SCHED %LOAD

Percentage value of the non-CPU-affined scheduling operations on the real CPU for an orderly distribution of the CPU capacity.



In the event of fixed CPU assignment (scheduling procedure “dedicated CPUs”), the value `DEDICATED SCHED` is output in the `NON-AFFIN SCHED` column.

Example of the VM-specific information block

SCHEDULE INFORMATION FOR VM(S)

CPU-POOL	VIRT	<--	TIME SLICE	-->	NON-AFFIN SCHED	%WAIT		
VM-ID	CPU	DEF	AVG %RUNOUT	%SHORT	%IDLE %LOAD	TIME		
*STDPPOOL								
1	MONITOR	00	8.00	0.24	1	62	DEDICATED SCHED	--
2	SYSTEM2	00	8.00	0.19	0	0	DEDICATED SCHED	--
EXCLPOOL								
4	SYSTEM4	00	99.00	VM-ACTIVE-IDLE			DEDICATED SCHED	--
		01		VM-ACTIVE-IDLE			DEDICATED SCHED	--

Meaning of the output columns

Percentage values in the VM-specific information block relate to the total of the values for all scheduling operations for a virtual CPU.

CPU-POOL

VM-ID

Identifies a CPU pool by means of its name or a VM by means of the VM index and VM name.

VIRT CPU

Virtual CPU identifier (CPU-ID).

TIME SLICE DEF

Current length of the time slice provided (see [page 252](#)).

TIME SLICE AVG

Average value of the time slices (in milliseconds) of the virtual CPU on a real CPU.

TIME SLICE %RUNOUT

Percentage value of the completely used time slices on the virtual CPU.

TIME SLICE %SHORT

Percentage value of the little-used time slices of the virtual CPU (value less than 0.1 milliseconds or less than a tenth of the time slice).



In the event of fixed CPU assignment (scheduling procedure “dedicated CPUs”), the value `VM-ACTIVE-IDLE` is output in the `<-- TIME SLICE -->` column if this parameter is effective for the VM.

NON-AFFIN SCHED %IDLE

Percentage value of the non-CPU-affined scheduling operations for the virtual CPU for avoiding unnecessary IDLE statuses of real CPUs.

NON-AFFIN SCHED %LOAD

Percentage value of the non-CPU-affined scheduling operations for the virtual CPU for an orderly distribution of the CPU capacity.



In the event of fixed CPU assignment (scheduling procedure “dedicated CPUs”), the value `DEDICATED SCHED` is output in the `NON-AFFIN SCHED` column.

%WAIT TIME

Wait time of the operable virtual CPU because no real CPU could be assigned (as a percentage of the length of the monitoring interval). Measurement for “delaying the VM”.

Example

One-off information output

See the detailed [“Example of CPU management” on page 253](#).

Periodic information output on a BS2000 console

```

/show-vm-status information=*all,period=60 _____ (1)
VMS3016 PERIODIC OUTPUT OF MONITORING DATA STARTED
VMS3000 VM2000 COMMAND /SHOW-VM-STATUS FOR VIRTUAL MACHINE ( ,*ALL)
        PROCESSED SUCCESSFULLY
NBR0740 COMMAND COMPLETED 'SHOW-VM-STATUS';
        (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
...
/SHOW-VM-STATUS DATA OUTPUT: (PERIOD=    60.003 SEC) _____ (2)
CPU TIMER INFORMATION FOR REAL CPU(S)
CPU-POOL  REAL      HPV-ACTIVE      HPV-IDLE  ALL-VMS-ACTIVE
          CPU       TIME/      %      TIME/      %      TIME/      %
*STDPOOL          3.401/ 2.83    107.121/89.26    9.484/ 7.90
... (output abbreviated)
... _____ (3)
/show-vm-status period=*no _____ (4)
/SHOW-VM-STATUS DATA OUTPUT: (PERIOD=    48.944 SEC)
CPU TIMER INFORMATION FOR REAL CPU(S)
CPU-POOL  REAL      HPV-ACTIVE      HPV-IDLE  ALL-VMS-ACTIVE
          CPU       TIME/      %      TIME/      %      TIME/      %
*STDPOOL          3.748/ 3.83    76.524/78.18    17.616/18.00
... (output abbreviated)
VMS3029 PERIODIC OUTPUT OF MONITORING DATA TERMINATED
VMS3000 VM2000 COMMAND /SHOW-VM-STATUS FOR VIRTUAL MACHINE ( ,*ALL)
        PROCESSED SUCCESSFULLY
NBR0740 COMMAND COMPLETED 'SHOW-VM-STATUS';
        (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>

```

- (1) The VM2000 administrator starts data collection for all VMs with a time period of 60 seconds.
- (2) The first measured values are output after approx. 60 seconds.
- (3) Further periodic outputs of monitored data follow.
- (4) Data collection is terminated. The remaining measured values are output.



The outputs are displayed asynchronously on a terminal. The output of the measured values is initiated only when an entry which is not empty (e.g. the /REMARK command) is made.

SHUTDOWN-VM

Execute a shutdown in the guest system and for VM2000

/SHUTDOWN-VM is used to execute a shutdown (without a restart) in the selected guest systems (as of BS2000/OSD V7.0).

The shutdown can be executed for the whole VM2000 system (all guest systems and the monitor system), for all guest systems or an individual guest system (with the exception of the monitor system).



Information on command execution is provided on [page 172](#).



On SQ servers the shutdown cannot be executed in a VM's guest system while the VM is being migrated (Live Migration).

On SQ servers a guest system can also be shut down via the SQ manager. After the shutdown the VM's firmware component continues to run until the VM is terminated. VM2000 operation can also be terminated (analogously to coordinated shutdown) via the SQ manager, see [section "Terminating the monitor system and VM2000" on page 79](#).

SHUTDOWN-VM

VM-IDENTIFICATION = *VM2000(...) / *VM-GUESTS / <integer 1..99> / <name 1..8>

*VM2000(...)

| **WAITING-TIME** = *UNLIMITED / <integer 1..60> *minutes*

,ACTION = *SHUTDOWN(...) / *ENTER-JOB

*SHUTDOWN(...)

| **QUIET-TIME** = <integer 0..60> *minutes*

VM-IDENTIFICATION =

Identification of the virtual machine for whose guest system the shutdown is to be executed.

VM-IDENTIFICATION = *VM2000(...)

The shutdown is to be coordinated and executed for the overall VM2000 system (with all guest systems). The shutdown is initiated for all guest systems (as of BS2000/OSD V7.0). The monitor system is the last system to be shut down. All VMs are also terminated.

- [S] No action is initiated for BS2000/OSD V6.0 guest systems. These should be shut down beforehand.
- [SQ] On SQ servers the coordinated shutdown for the entire VM2000 system cannot be executed while a VM is being migrated (Live Migration).

WAITING-TIME =

Determines the maximum time which is to be waited (before the monitor system is terminated) for all virtual machines to have the status INIT-ONLY or DOWN. The wait time begins with the start of the enter jobs or when shutdown processing is initiated (MODE=*END) in the guest systems. In this period the monitor system continues to operate regularly. VM2000 commands can also still be entered, see [page 171](#).



The maximum wait time selected should be greater than the longest SHUTCTL wait time in the guest systems (as of BS2000/OSD V7.0), see system parameter SHUTCTL in the “Introductory Guide to Systems Supports” [2]. The guest systems can consequently be shut down before the monitor system.

WAITING-TIME = *UNLIMITED

The monitor system waits for the other guest systems to shut down for an unlimited period before it shuts down itself.



This operand value is rejected if a VM is in the IN HOLD (WAIT) status.

- [S] This operand value is rejected if guest systems BS2000/OSD V6.0 are running.

WAITING-TIME = <integer 1..60> minutes

Time in minutes during which the guest system remains in QUIET status before it is shut down.

All guest systems (with the exception of the monitor system) are explicitly shut down after the wait time has elapsed using /DELETE-VM CHECK-VM-STATE=*NO regardless of their status.

VM-IDENTIFICATION = *VM-GUESTS

The shutdown is to be executed for all guest systems (as of BS2000/OSD V7.0) with the exception of the monitor system.

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>

Identifies the virtual machine by means of a VM index or VM name. The guest system must be a guest system BS2000/OSD as of V7.0.

ACTION =

Defines the method for system shutdown of a guest system.

ACTION = *SHUTDOWN(...)

The system shutdown of a guest system is implemented by VM2000 by calling shutdown processing (analogously to the BS2000 command /SHUTDOWN). The operator on the guest system is informed of this by the message NRT1201.

QUIET-TIME =

Time in minutes during which the guest system remains in QUIET status before it is shut down.

QUIET-TIME = 0

System shutdown is initiated immediately.

QUIET-TIME = <integer 1..60> minutes

System shutdown is executed in two steps:

1. The system is initially placed in QUIET status:
 - All interactive users of the system are informed of the impending system shutdown
 - Job acceptance is terminated
 - New users are no longer admitted

The guest system remains in this status for the defined QUIET-TIME. The operator on the guest system is informed of the length of the QUIET-TIME.

2. After the QUIET-TIME has elapsed, the final system shutdown is initiated.

ACTION = *ENTER-JOB

The system shutdown of a guest system is to be implemented by an enter job. The command file is expected in the \$TSOS.SYS.VM.SHUTDOWN file in the guest system. The last action of the customer specific preparations must be the BS2000 command /SHUTDOWN MODE=*NQ / *END(RESTART=*NQ). The privilege OPERATING is required for the execution of /SHUTDOWN (see [page 92](#)).

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
	0	CMD0001	Shutdown successfully initiated
1	0	CMD0001	No action required
2	0	VMS2104	VM2000 is shut down; at least one VM is in the IN HOLD (WAIT) status
	64	VMS2055	Shutdown of guest system not acknowledged
	64	VMS2056	Shutdown for guest system already in progress
	64	VMS2057	Shutdown for VM rejected; no suitable guest system is running
	64	VMS2058	Shutdown of guest system rejected or acknowledged with warnings
	64	VMS2070	VM in the IN HOLD (WAIT) state
	64	VMS2101	VM2000 shutdown in progress
	64	VMS2102	Shutdown for VM2000 rejected; monitor system unsuitable
	64	VMS2103	Unlimited waiting for shutdown of an unsuitable guest system not permitted
	64	VMS2105	/ENTER-JOB cannot be executed in the guest system
	64	VMS4000	VM not yet created
	64	VMS9005	Command could not be processed or could only be partially processed (VM-ID=*VM-GUESTS)
	128	VMS3042	VM being migrated. Command rejected

Guaranteed messages

VMS2054, VMS2055, VMS2056, VMS2057, VMS2058, VMS2070, VMS2100, VMS2104, VMS3041

Example

```
/shutdown-vm vm-id=*vm2000(waiting-time=*unlimited),
                    action=*shutdown(quiet-time=2) _____ (1)
VMS2054 SHUTDOWN OF GUEST SYSTEM ON VM (2,SYSTEM2) INITIATED
VMS2054 SHUTDOWN OF GUEST SYSTEM ON VM (3,SYSTEM3) INITIATED
VMS2100 VM2000 SHUTDOWN INITIATED
```

- (1) The VM2000 administrator initiates the coordinated shutdown of VM2000. The monitor system would wait for an unlimited time for the guest systems to shut down. The QUIET-TIME for the guest systems and the monitor system is two minutes.

See also the detailed [“Example of coordinated shutdown of VM2000” on page 81](#).

START-VM

Start a guest system on a VM

/START-VM performs initial program loading for the guest system on a VM. After command execution, the VM is in one of the states RUNNING (operand HOLD=*NO) or IN HOLD (WAIT) (operand HOLD=*YES, S servers).

[SQ] On SQ servers the guest system can also be started via the SQ manager or using the SVP functions of the KVP console which is assigned to the VM (see the manual “SQ Series Business Servers, Operation and Administration” [18]).

The command cannot be issued for the monitor VM. The [section “Initializing VM2000 operation” on page 65](#) describes how to start the monitor system.

i /START-VM presupposes that the VM has already been assigned the resources and devices required during startup. Furthermore, the resources must be activated and the channels must be attached. There must be at least one real, attached CPU in the CPU pool to which the VM is assigned.

START-VM

```

IPL-UNIT = *STD / <alphanum-name 2..2> / <x-text 4..4>
,MAIN-CONSOLE = *STD / list-poss(2): <alphanum-name 2..2>
,HOLD = *NO / *YES
,DIAGNOSTIC-IPL = *NO / *YES
,INFORMATION-BYTE = *FAST / *AUTOMATIC / *DIALOG
,PARAMS = *NONE / <alphanum-name 1..8>
,CLEAR-MEMORY = *NO / *YES
,UNLOCK-SAVEAREA = *NO / *YES
,VM-IDENTIFICATION = *CURRENT / <integer 1..99> / <name 1..8>
,CHECK-VM-STATE=*YES / *NO

```

IPL-UNIT =

Mnemonic device name of the disk device used for the initial program loading (IPL) of the guest system. The device must be assigned to the VM explicitly and exclusively.

[S] If there is no KVP console device pair assigned to the VM on S servers, the virtual console of the VM is used.

IPL-UNIT = ***STD**

The IPL device used at the last start (restart) of this VM is used again. When a VM is first started, the IPL device must be specified explicitly.

IPL-UNIT = <alphanum-name 2..2> / <x-text 4..4>

Explicitly specifies the device used for the IPL of the guest system (2- or 4-digit mn).

MAIN-CONSOLE =

Specifies the main console for the guest system.

MAIN-CONSOLE = *STD

The operand entry of the last IPL is used. At the first /START-VM, VM2000 looks for a KVP console console device pair assigned to the VM.

- [S] If there is no KVP console console device pair assigned to the VM on S servers, the virtual console of the VM is used.

MAIN-CONSOLE = list-poss(2): <alphanum-name 2..2>

The pair of mnemonic device names (Output path, Input path) for a KVP console device pair must be specified in this order.

- [S] For S servers, the mnemonic device name of the virtual console can also be entered.

HOLD = *NO / *YES

This determines the state of the VM after initial program loading of the guest system and generation of its context.

- [SQ] Only the default value may be used on SQ servers.

HOLD = *NO

The VM is in the RUNNING state.

HOLD = *YES

The VM is in the IN HOLD (WAIT) state.

DIAGNOSTIC-IPL = *NO / *YES

Specifies whether a diagnostic IPL or a normal IPL is to be carried out.



Implicitly assigned devices are removed from the VM for a normal IPL (DIAGNOSTIC-IPL=*NO). In the case of a diagnostic IPL (DIAGNOSTIC-IPL=*YES), implicitly assigned disk devices remain with the VM for diagnostic reasons. Exclusive use is changed to shared use. Other, implicitly assigned devices are released.

INFORMATION-BYTE = *FAST / *AUTOMATIC / *DIALOG

One byte of load information for the guest system.

During normal initial program loading (DIAGNOSTIC-IPL=*NO), this defines the type of system startup (fast, dialog or automatic startup).

During the diagnostic IPL (DIAGNOSTIC-IPL=*YES), this defines the SLED execution type: manual (*FAST, *DIALOG) or automatic (*AUTOMATIC).

[SQ] On SQ servers the execution type of the SLED is always set manually in the case of a diagnostic IPL. The values of the `INFORMATION-BYTE` operand is ignored.

PARAMS = *NONE / <alphanum-name 1..8>

Transfer information for the guest system.

[SQ] Only the default value may be used on SQ servers.

The information corresponds to the specification in the `PARMS` field of the corresponding SVP menu screen (see [section “Initializing VM2000 operation” on page 65](#)).

CLEAR-MEMORY = *NO / *YES

Specifies whether or not the main memory of the VM is to be deleted for an IPL. This operand is only evaluated when `DIAGNOSTIC-IPL=*NO` is specified.

UNLOCK-SAVEAREA = *NO / *YES

Specifies whether, in the event of a diagnostic IPL, the save area is to be overwritten (SLED of SLED) or not. The operand is only evaluated when `DIAGNOSTIC-IPL=*YES` is specified.

VM-IDENTIFICATION =

Identification of the virtual machine.

VM-IDENTIFICATION = *CURRENT

The VM identification assigned previously by means of `/BEGIN-VM-DIALOG` is used.

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>

Identifies the virtual machine by means of a VM index or VM name.

CHECK-VM-STATE =

Determines whether the command is executed, depending on the current state of the VM.

CHECK-VM-STATE = *YES

The command is only executed if no guest system is running on the VM. This is the case if the VM is in the `DOWN` or `INIT-ONLY` state.

[SQ] On SQ servers a VM cannot be started with `CHECK-VM-STATE=*YES` while it is being migrated (Live Migration).

CHECK-VM-STATE = *NO

The command is executed regardless of the VM state. The guest system running on the VM is halted without regard for the current system state. Files of the guest system could be inconsistent as a result.

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
2	0	VMS2020	VM remains in IN HOLD(WAIT) state
	64	VMS2101	VM2000 shutdown in progress
	64	VMS3002	Invalid operand value
	64	VMS3040	Not processed because of the state of the VM
	64	VMS4000	VM not yet created
	64	VMS4016	Device cannot be used as a virtual console
	64	VMS4027	No console is assigned to the virtual machine
	64	VMS4034	IPL device currently shared among virtual machines
	64	VMS4037	I/O error when reading BOOT
	64	VMS4038	Timeout when starting the VM's firmware component or the IPL CPU
	64	VMS4050	RSC could not be stopped
	64	VMS4080	Device could not be reset
	64	VMS4084	I/O configuration update in progress
	64	VMS4160	Device pair in MAIN-CONSOLE is not a KVP console
	64	VMS4161	The first device of the KVP console is not an output path
	64	VMS4612	No attached CPU in the CPU pool
	64	VMS9006	Specified hardware devices not suitable or not available
	128	VMS3042	VM being migrated. Command rejected

Example

```
/start-vm ipl-unit=d2,information-byte=*dialog
```

The VM administrator starts the guest system on his/her VM. The disk device assigned via the mnemonic device name D2 is the IPL disk. A dialog startup is requested. A dialog startup is to be performed.

SWITCH-VM-CPU

Delete CPUs from a CPU pool and add them to another CPU pool

/SWITCH-VM-CPU is used to delete a CPU from a CPU pool and add it to a different CPU pool.

This command is reserved for the VM2000 administrator.

SWITCH-VM-CPU

CPU-IDENTIFICATION = *SELECT(...) / ***ALL** / list-poss(63): <x-text 1..2>

***SELECT(...)**

NUMBER-OF-CPUS = 1 / <integer 1..63> / ***ALL**

,ATTRIBUTE = *NORMAL / ***EXTRA**

,STATE = *ATTACHED / ***DETACHED** / ***OFFLINE**

,FROM-CPU-POOL = *STDPOOL / <name 1..8>

,TO-CPU-POOL = *STDPOOL / <name 1..8>

,ATTACHED-VM-CPUS = *ACTIONS(...)

***ACTIONS(...)**

SOURCE-CPU-POOL = *CHECK-NUMBER / ***ADJUST-NUMBER**

,TARGET-CPU-POOL = *NONE / ***ADJUST-NUMBER**

CPU-IDENTIFICATION =

Determines the CPUs which are to be switched from the source CPU pool (**FROM-CPU-POOL** operand) to the target CPU pool (**TO-CPU-POOL** operand).

CPU-IDENTIFICATION = *SELECT(...)

Selects a set of CPUs from the source CPU pool (**FROM-CPU-POOL** operand).

NUMBER-OF-CPUS = 1 / <integer 1..63> / ***ALL**

Number of CPUs.

ATTRIBUTE = *NORMAL / ***EXTRA**

CPU type.

STATE = *ATTACHED / ***DETACHED** / ***OFFLINE**

Reconfiguration status of the CPUs.

CPU-IDENTIFICATION = *ALL

All CPUs in the source CPU pool are selected.

CPU-IDENTIFICATION = list-poss(63): <x-text 1..2>

Selects the specified real CPUs.

FROM-CPU-POOL = *STDPOOL / <name 1..8>

Determines the source CPU pool from which the CPUs are to be deleted.

TO-CPU-POOL = *STDPOOL / <name 1..8>

Determines the target CPU pool to which the selected CPUs are to be added.

ATTACHED-VM-CPUS = *ACTIONS(...)

Determines the command execution and the implicit actions of VM2000.

SOURCE-CPU-POOL =

Determines the command execution and the implicit actions of VM2000 for the virtual CPUs of the VMs which are assigned to the source CPU pool.

The command is rejected if the last attached real CPU is to be deleted, but a VM is still assigned to the source CPU pool (on S servers in RUNNING or IN HOLD(WAIT) state) (message/return code VMS4612).

SOURCE-CPU-POOL = *CHECK-NUMBER

The command is executed if the number of attached virtual CPUs of each VM in the source CPU pool is less than or equal to the remaining number of attached real CPUs in the source CPU pool, see [page 228](#). Otherwise the command is rejected. Excess virtual CPUs which have already been detached can then not be attached again to the guest system (BLOCK status).

SOURCE-CPU-POOL = *ADJUST-NUMBER

The command is executed.

Here VM2000 checks the status of the virtual CPUs of **all** VMs in the source CPU pool and, if required, initiates the following asynchronous actions:

- If the number of attached virtual CPUs of a VM is greater than the number of attached real CPUs in the source CPU pool, VM2000 detaches the excess virtual CPUs of the VMs concerned (implicit /DETACH-DEVICE in the BS2000 guest system for a corresponding number of virtual CPUs). These detached CPUs cannot initially be attached again by the guest system; they are identified accordingly by VM2000 (status of the virtual CPU in the VM2000 information commands: BLOCK).
- If the number of attached virtual CPUs of a VM is less than the number of attached real CPUs of the source CPU pool (e.g. through CPU reconfiguration), VM2000 attaches the detached virtual CPUs of the VMs concerned (implicit /ATTACH-DEVICE in the BS2000 guest system for a corresponding number of virtual CPUs).

TARGET-CPU-POOL =

Determines the implicit actions of VM2000 for the virtual CPUs of the VMs which are assigned to the target CPU pool.



For all VMs whose number of attached virtual CPUs is less than their multiprocessor level, VM2000 cancels the BLOCK status for the relevant number of virtual CPUs in the VM. The virtual CPU can then be attached again in the guest system.

TARGET-CPU-POOL = *NONE

No further implicit actions are performed. The number of attached virtual CPUs in the VMs remains unchanged.

TARGET-CPU-POOL = *ADJUST-NUMBER

VM2000 checks the status of the virtual CPUs of **all** VMs in the target CPU pool and, if required, initiates the following asynchronous actions:

- If the number of attached virtual CPUs of a VM is less than the (new) number of attached real CPUs of the target CPU pool, VM2000 attaches the detached virtual CPUs of the VMs concerned (implicit /ATTACH-DEVICE in the BS2000 guest system for a corresponding number of virtual CPUs).
- If the number of attached virtual CPUs of a VM is greater than the (new) number of attached real CPUs of the target CPU pool (e.g. through CPU reconfiguration), VM2000 detaches the excess virtual CPUs of the VMs concerned (implicit /DETACH-DEVICE in the BS2000 guest system for a corresponding number of virtual CPUs). These detached CPUs cannot initially be attached again by the guest system; they are identified accordingly by VM2000 (status of the virtual CPU in the VM2000 information commands: BLOCK).

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
2	0	VMS3004	Multiple occurrence of elements in CPU list
2	0	VMS4603	No CPU belongs to the CPU pool
	64	VMS3002	Invalid operand value (CPU-IDENTIFICATION)
	64	VMS3003	Invalid combination of operand values (FROM-CPU-POOL = TO-CPU-POOL)
	64	VMS4043	CPU not known in system
	64	VMS4600	CPU pool not yet created
	64	VMS4606	CPU switch rejected; command aborted
	64	VMS4607	CPU does not belong to source CPU pool
	64	VMS4611	Selected CPUs do not belong to source CPU pool
	64	VMS4612	Too few attached CPUs remaining in source CPU pool
	64	VMS9004	Functionality not supported on current HSI or with current parameter assignment

Guaranteed messages

VMS4608, VMS4620, VMS4621

Example

```

/switch-vm-cpu cpu-id=(02,03),from-cpu-pool=*stdpool,to-cpu-pool=pcust1,
  attached-vm-cpus=*actions(source-cpu-pool=*adjust-number) _____ (1)
VMS4608 CPU '02' SWITCHED FROM CPU-POOL '*STDPOOL' TO CPU-POOL 'PCUST1'
VMS4608 CPU '03' SWITCHED FROM CPU-POOL '*STDPOOL' TO CPU-POOL 'PCUST1'
VMS4621 IMPLICIT DETACH-DEVICE FOR VIRTUAL CPUS ON
      VIRTUAL MACHINE (2,C1SYS1) STARTED _____ (2)
% VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (2,C1SYS1),
      CODE 80FFFFFF, CPU 02
% VMS2011 CPU 02 OF VIRTUAL MACHINE (2,C1SYS1) STOPPED

```

- (1) The real CPUs 02 and 03 are switched from the standard CPU pool to the CPU pool PCUST1.
- (2) The SOURCE-CPU-POOL=*ADJUST-NUMBER operand causes VM2000 to check, for all VMs in the source CPU pool, whether the number of virtual CPUs which they possess remains greater than or equal to the remaining number of real CPUs. In this case VM2000 initiates a detach operation for virtual CPU 02 in the customer system as only two real CPUs are left in the standard CPU pool but three virtual CPUs are running in VM C1SYS1.

See also the detailed [“Example of managing CPU pools” on page 233](#).

SWITCH-VM-DEVICES

Deallocate devices from one VM, allocate them to another

`/SWITCH-VM-DEVICES` deallocates devices from one VM and allocates them to another VM. Individual devices are addressed using their mnemonic device names. The devices of a pubset are determined by means of the catalog ID of the pubset or the mnemonic device name of the pubres device.

This command is reserved for the VM2000 administrator.

If `/ADD-VM-DEVICES` was used to define a device attribute (using the `TYPE` operand), this will be retained when `/SWITCH-VM-DEVICES` is used. For implicitly assigned devices the usage mode is preserved (`EXCLUSIVE/SHARED`).

The home pubset and other imported pubsets, paging disks and the final console cannot be removed from the monitor VM.



The devices removed from a VM should be disconnected beforehand in the corresponding guest system with `/DETACH-DEVICE`.

The devices must be attached in the guest system to which they are assigned so that they can be addressed in it.



Alias devices (PAV) cannot be explicitly removed.

SWITCH-VM-DEVICES

```

UNITS = *ALL(...) / *PUBSET-DEVICES(...) /
        list-poss(28): *DEVICE-RANGE(...) / <alphanum-name 1..4 with-wild(20)>

*ALL(...)
  | CHECK-VM-STATE = *YES / *NO
*PUBSET-DEVICES(...)
  | PUBSET = *BY-PUBRES-DEVICE(...) / <cat-id 1..4>
    | *BY-PUBRES-DEVICE(...)
      | UNIT = <alphanum-name 2..2> / <x-text 4..4>
*DEVICE-RANGE(...)
  | FROM = <alphanum-name 2..2> / <x-text 4..4>
    | ,TO = <alphanum-name 2..2> / <x-text 4..4>
,FROM-VM = <integer 1..99> / <name 1..8>
,TO-VM = <integer 1..99> / <name 1..8>
,FORCE = *NO / *YES

```

UNITS =

Specifies the devices to be removed from the FROM-VM and assigned to the TO-VM.

UNITS = *ALL(...)

All devices assigned to the FROM-VM are to be removed and reassigned. If a device cannot be removed, the command is not executed for this device. Alias devices are only removed (implicitly).

CHECK-VM-STATE =

Determines whether the command is executed depending on the current status of the FROM-VM.

CHECK-VM-STATE = *YES

The command is only executed if the FROM-VM is in the DOWN or INIT-ONLY state.

CHECK-VM-STATE = *NO

The command is executed regardless of the FROM-VM state.

UNITS = *PUBSET-DEVICES(...)

All devices are to be removed and reassigned.



See also [“Notes on pubset specification in the VM2000 commands” on page 313](#) and [“Pubset reconfiguration” on page 315](#).

PUBSET = *BY-PUBRES-DEVICE(...)

Determines the devices of the pubset by means of the pubres device.

UNIT = <alphanum-name 2..2> / <x-text 4..4>

Mnemonic device name of the pubres device.

PUBSET = <cat-id 1..4>

Catalog ID of the pubset. The pubset must be entered together with its pubres device in the monitor system's MRS catalog.

UNITS = *DEVICE-RANGE(...)

All devices in this name range are to be removed and reassigned.



Device names <x-text 4..4> (in *DEVICE-RANGE) are selected from FROM to TO in hexadecimal order. Otherwise, device names <alphanum-name 2..2> are selected in alphabetical order.

If UNITS is specified with *DEVICE-RANGE or with wildcards, alias devices and devices that are not assigned to the FROM-VM are ignored.

FROM = <alphanum-name 2..2> / <x-text 4..4>

Defines the start of the name range.

TO = <alphanum-name 2..2> / <x-text 4..4>

Defines the end of the name range. The mnemonic device name must be in the same value range (2- or 4-digit mn) as the FROM device name and must be greater than or equal to the FROM device name in ascending order in the value range.

UNITS = <alphanum-name 1..4 with-wild(20)>

Specifies the devices whose assignment is to be output.

FROM-VM = <integer 1..99> / <name 1..8>

Identifies by means of a VM index or VM name the VM from which the devices are to be removed.

TO-VM = <integer 1..99> / <name 1..8>

Identifies by means of a VM index or VM name the VM to which the devices are to be assigned.

FORCE =

Specifies the conditions under which devices are to be removed from the FROM-VM.

FORCE = *NO

The device is removed from the FROM-VM if the FROM-VM is in the DOWN or INIT-ONLY state, or if the device is detached in the guest system. However, it is not removed if an input/output is outstanding for the device, if the FROM-VM is in the wait state (after /HOLD-VM) or if the guest system is in the startup phase.

[SQ] On SQ servers the device configuration of a VM cannot be modified with FORCE=*NO while it is being migrated (Live Migration).

FORCE = *YES

VM2000 tries to remove the device from the FROM-VM, even if it is still in use. If this is not possible, message VMS4003 or VMS4031 is output. The operand should only be used in exceptional circumstances.

[S] The IO-RESET privilege of the FROM-VM must be noted on S servers (see [page 145](#)).

Command return code, see also [page 423](#)

(SC2)	SC1	Maincode	Meaning
	0	CMD0001	Command successfully executed
1	0	VMS4035	No corresponding device found
1	0	VMS4180	Specified set of devices empty
2	0	VMS4229	Not all devices of the pubset could be processed
	64	VMS3002	Invalid operand value (e.g. *DEVICE=RANGE operand) or FROM-VM=TO-VM
	64	VMS3040	Not processed because of the state of the VM
	64	VMS4000	VM not yet created
	64	VMS4006	Device is not available
	64	VMS4007	Device is not assigned to the virtual machine
	64	VMS4021	Device is the last console or main console of the monitor system. Reconfiguration not possible
	64	VMS4022	Device is currently being used as a public disk in the monitor system. Reconfiguration not possible
	64	VMS4026	VM already has a virtual console
	64	VMS4030	The monitor VM may not be assigned a virtual console
	64	VMS4031	Device is INOP
	64	VMS4064	No channel path available
	64	VMS4069	Device is already assigned to the TO-VM
	64	VMS4081	Device only removed. Switching incomplete
	64	VMS4085	Device not detached in guest system (FORCE=*NO)
	64	VMS4086	Device state not known in the guest system (FORCE=*NO)
	64	VMS4143	Guest system is in the startup phase (FORCE=*NO)
	64	VMS4144	VM is in the wait state (FORCE=*NO)
	64	VMS4190	Device is an alias device (PAV)
	64	VMS4223	Command not executed (operand *PUBSET-DEVICES)
	64	VMS9005	Command could not be processed or could only be partially processed
	128	VMS3042	VM being migrated. Command rejected
	128	VMS4003	Device is busy

Guaranteed messages

VMS4003, VMS4008, VMS4021, VMS4022, VMS4026, VMS4030, VMS4031, VMS4064, VMS4069, VMS4081, VMS4085, VMS4086

Only if *PUBSET-DEVICES is specified: VMS4223, VMS4228, VMS4229.

Only if UNITS is specified without *DEVICE=RANGE and without with wildcards: VMS4006, VMS4007, VMS4190.

Example

```
/switch-vm-devices from-vm=system2,to-vm=system3,  
    units=*pubset-devices(pubset=*by-pubres-device(unit=d2)) ----- (1)  
VMS4008 DEVICE 'D2' SWITCHED FROM VIRTUAL MACHINE (2,SYSTEM2)  
        TO VIRTUAL MACHINE (3,SYSTEM3) ----- (2)  
VMS4228 PUBSET 'VM02' WITH PUBRES DEVICE 'D2' SWITCHED  
        FROM VM (2,SYSTEM2) TO VM (3,SYSTEM3)
```

- (1) The VM2000 administrator removes the device(s) of the pubset with the pubres device `D2` from VM2 and assigns them to VM3.
- (2) VM2000 confirms the old and new assignments.

Glossary

Only terms specific to VM2000 are included in this glossary. It is assumed that the user is already familiar with terms from other operating systems used, from the hardware-software interface (HSI) and BS2000/OSD.

Cross-references to other terms are printed in *italics*.

ADMIN dialog

Synonym for the administration dialog of the *VM administrator* for operating his/her VM or of the *VM2000 administrator* for operating *VM2000* via a privileged user task or via *\$VMCONS*.

assignment mode

Type of assignment of a device or the devices of a pubset under VM2000:

- usable by only one VM (*EXCLUSIVE*)
- usable by multiple VMs (*SHARED*) with *direct (SH(D))* or *indirect (SH(I))* I/O processing

assignment set

Assignment sets are groups of devices with the *ASSIGN-BY-GUEST* attribute which, for organizational reasons, are regarded as a unit for implicit device assignment to VMs. Each device which is intended for implicit assignment must have the *ASSIGN-BY-GUEST* attribute and be assigned to precisely one assignment set.

BS2000 VM

Virtual machine with a BS2000/OSD guest system.

CPU pool

Combination of real normal or extra CPUs to form an organizational unit.

CPU share

The CPU share of a VM is the share of the CPU capacity of the *CPU pool* which the VM may use or is using in a time period.

CPU share, current

Planned value for the *CPU share* of the *CPU pool's* CPU capacity which a VM has, using the current values of the active objects for VM2000 operation. Comparison value for measurements under VM2000.

CPU share, effective

Planned value for the *CPU share* of the *CPU pool's* CPU capacity which a VM has under idealized conditions.

guest system

Operating system which runs on a *virtual machine*.

HSI

Hardware-software interface. Depending on the application, the term is defined in detail as: *real HSI*, *virtual HSI*, *VM2000 HSI*.

hypervisor (VM2000 hypervisor)

Part of the *VM2000* on S servers which controls the execution of *guest systems* on *virtual machines*. On SQ servers this function is executed by the Xen hypervisor.

hypervisor mode (VM2000 hypervisor mode)

Processor mode in *VM2000-HSI* on S servers with new registers and commands in which the *VM2000 hypervisor* runs.

monitor (VM2000 monitor)

Part of *VM2000* via which *virtual machines* are operated by means of *VM2000 commands*.

monitor system

Preferred *guest system* on which the *monitor* is located. It is the guest system on the *monitor VM*.

monitor VM

Virtual machine on which the *monitor system* runs.

multiprocessor level

Mnemonic expression for the number of virtual CPUs of a VM.

OSD/XC

OSD Extended Configuration (OSD/XC) is the BS2000/OSD operating system for the SQ servers whose functionality is complemented by additional system-related software products.

real HSI

Hardware-software interface of the BS2000/OSD server which has not been supplemented by firmware components.

scheduling

When a VM is scheduled under VM2000 on S servers, the VM2000 *hypervisor* starts an operable virtual CPU of a VM on a free real CPU. On SQ servers this function is executed by the Xen hypervisor.

usage mode

Type of usage of a device or the devices of a pubset under VM2000:

- explicit (by the VM2000 or VM administrator)
- implicit (by the guest system)

VC dialog

Synonym for the dialog of the *guest system* operator via *\$VMCONS* for operating his/her *guest system*. On S servers the *virtual console* is required for this.

virtual console

Device on S servers, which is generated as a physical console in both the *monitor system* and the *guest system*, but which need not be physically present. If the virtual console is used as the *guest system* console, the exchange of messages is handled as a *VC dialog* via *\$VMCONS*. The virtual console is not used on SQ servers.

virtual CPU (virtual processor)

CPU of a *virtual machine*.

virtual HSI

Hardware-software interface of a *virtual machine*.

virtual machine system

Program system which offers several identical or different *virtual machines* on a real server.

virtual machine, VM

A virtual BS2000/OSD server for the user and on which a *guest system* runs.

VM

Abbreviation for *virtual machine*.

VM administrator

Administrator of a *VM*. He/she may issue only *VM administrator commands* (*ADMIN dialog*).

VM administrator command

This is a command for use by the *VM administrator* and controls operation of a *VM*.

VM group

On S servers, combination of VMs to form an organizational unit with its own attributes.

VM identification, VM-ID

Identification of a VM. It can be the *VM index* or the *VM name*.

VM index

Integer $n \geq 1$; identifies the *VM* as VM1 to VMn.

VM mode

Processor mode in *VM2000-HSI* on S servers in which the *virtual machines* run.

VM name

Name of a *VM*; identifies the VM in addition to the *VM index*.

VM2000

Name of the *virtual machine system* described here.

VM2000 administrator

Administrator of the *monitor VM*. Only he/she may input *VM2000 administrator commands* (*ADMIN dialog*).

VM2000 administrator command

Command for use by the *VM2000 administrator* for controlling the overall *VM2000* system.

VM2000 command

Generic term for *VM2000 administrator command* and *VM administrator command*.

VM2000 HSI

HSI which is offered by the BS2000/OSD server for an operating system (HSI of a partition or *real HSI*) on S servers, including VM2000 extensions in the form of firmware components. The VM2000 HSI on S servers offers two processor modes, (*VM2000*) *hypervisor mode* and *VM mode*.

\$VMCONS

BCAM application in the *monitor system*; it controls and monitors the exchange of messages for operating the *VMs* (*ADMIN dialog*) and for operating the *guest systems* (*VC dialog*).

VMn

Synonym for the *VM* with the *VM index* n.

XenVM

Virtual machine on SQ servers with a Linux or Windows guest system.

Related publications

You will find the manuals on the internet at <http://manuals.ts.fujitsu.com>. You can order manuals which are also available in printed form at <http://manualshop.ts.fujitsu.com>.

- [1] **BS2000/OSD-BC
System Installation**
User Guide
- [2] **BS2000/OSD-BC
Introductory Guide to Systems Support**
User Guide
- [3] **BS2000/OSD-BC
Utility Routines**
User Guide
- [4] **BS2000/OSD-BC
Diagnostics Handbook**
User Guide
- [5] **DRV (BS2000/OSD)
Dual Recording by Volume**
User Guide
- [6] **ELSA (BS2000/OSD)
Error Logging System Analysis**
User Guide
- [7] **RAV (BS2000/OSD)
Computing Center Accounting Procedure**
User Guide
- [8] **openSM2 (BS2000/OSD)
Software Monitor**
User Guide

- [9] **HIPLEX MSCF** (BS2000/OSD)
BS2000-Processor Networks
User Guide
- [10] **SECOS** (BS2000/OSD)
Security Control System - Access Control
User Guide
- [11] **OMNIS/OMNIS-MENU** (BS2000/OSD)
Functions and Commands
User Guide
- [12] **BS2000/OSD-BC**
Commands
User Guides
- [13] **IMON** (BS2000/OSD)
Installation Monitor
User Guide
- [14] **PROP-XT** (BS2000/OSD)
Programmed Operating with SDF-P
Product Manual
- [15] **BS2000/OSD-BC**
Executive Macros
User Guide
- [16] **JV** (BS2000/OSD)
Job Variables
User Guide
- [17] **BS2000/OSD**
Performance Handbook
User Guide
- [18] **SQ Series Business Servers**
Operation and Administration
User Guide
- [19] **SHC-OSD / SCCA-BS2**
Storage Management for BS2000/OSD
User Guide

- [20] **SKP 3970** (for SKP 3970-40 and -50)
Service and Console Processor
User Guide
- SKP 3970-2x**
Service and Console Processor
User Guide
- SINIX-2000** (for SKP 3970-20)
Installation, Operation and Administration
User Guide
- [21] **HNC**
High-Speed Network Connect
User Guide

Index

\$CONSOLE 101, 110
\$TSOS.SYS.VM.SHUTDOWN 170, 625
\$VMCONS, see VMCONS

3970 SCP 102

A

absolute address 265
accounting 20
 device 286
 global storage 281
 in VM2000 380
 main memory 265
 pubset 306
accounting procedure 211
accounting record
 for a VM 139, 381, 489
 for devices 381, 384
 structure 383
ADD-VM-DEVICES 428
 for the monitor VM 58, 72
ADD-VM-RESOURCES 433
 for the monitor VM 58, 72
ADD-VM-TO-VM-GROUP 154, 179, 436
adding VM to VM group 436
additional mirror unit 319
addressing pubset 305
ADMIN dialog 89, 155, 641
 example 95
 via \$VMCONS 115
 via user task 91
administration application \$VMCONS
 see VMCONS
administration dialog 89

agent, VM2000 31
alias device (PAV) 345, 495
allocating devices to a VM 285, 428
analysis program
 DAMP 412
 VMDUMP 406
application availability 21
application pool 34
ASSIGN-BY-GUEST (operand) 459, 488
ASSIGN-VM-GROUP-TO-CPU-POOL 154, 232, 438
ASSIGN-VM-TO-CPU-POOL 154, 232, 440
assigning
 devices to a VM 154
 global storage 433
 pubsets to a VM 154, 307
 VM group to CPU pool 438
 VM to a CPU pool 440, 452
 VM1 to a CPU pool 57
assignment mode 641, 643
assignment set 154, 161, 288, 308, 641
asynchronous messages 93
ATTACH-DEVICE 197, 219, 262
ATTACH-VM-RESOURCES 218, 442
attaching, real CPU 442
attributes of a VM 160
 assigning 138, 452
 modifying 161
attributes of a VM group
 assigning 178, 465
 modifying 499
authorization code 100, 101
AUTO-SNAP-ASSIGNMENT (operand) 458, 488
automatic restart of the monitor system 402
automatic restart of VM2000 401

AUTOMATIC-STARTUP 59, 69

availability cluster 21

B

backup file for VM2000 configuration 367

example 372

location 367

Structure of the contents 367

XML document 367

base device 345

assigning to VM 347

changes 347

usage mode 347, 495

BCAM line connection 353

BCV 319

BEGIN-VM-DIALOG 92, 115, 118, 155, 446

BI (multiprocessor level) 456

Boolean (data type of the S variables) 425

BS2000 console 100

BS2000 device 34, 47

BS2000 procedures 130

BS2000 VM 641

Business Continuance Volume 319

C

CALL-VM-PROCEDURE 91, 449

cancel the wait state of a VM 523

capacity increasing 222

Capacity on Demand 222

carrier system 30

changing

usage mode of a device 292, 495

usage mode of a pubset 311

channel 359

channel reconfiguration 263

Channel Subsystem Monitoring Facility 39, 392, 592

class-2 system parameter

SHUTARCH 172

clone unit 319

color control 118, 122

combining operating levels (\$VMCONS) 114

command range 138, 144, 452

extend 192, 458, 487, 489

modifying 484

restrict 192, 458, 487

command return code 423

command sequences 130

command/function range for the VM2000

administrator 57

commands of VMHEL 390

communication devices 336

configuration

in X2000 62

of peripherals 45

redundant 45

configuration file

for the monitor system 56, 69, 71, 72

configuration notes 358

for HNC 352

configuration properties, device 201

configuring

monitor system 48

conflict 377

connection name 112, 114

CONSLOG file 93, 116

console

BS2000 100

logical 101

virtual, see virtual console

console distribution program 102, 107

constraints 37

guest system 40

hardware 38, 39

software 39

VM2000 39

contents of the S variables 425

context of a VM 29

continuation lines 449

controller 359

coordinated shutdown 80, 171

CPU

real 213

virtual 215

CPU capacity parameters 238

CPU intensity 251

- CPU pool [20](#), [34](#), [138](#), [143](#), [178](#), [179](#), [452](#), [641](#)
 - adding a VM [225](#)
 - adding a VM group [225](#)
 - adding CPU [225](#), [232](#)
 - and scheduling [250](#), [251](#)
 - assigning VM [232](#)
 - creating [231](#)
 - example [233](#)
 - information [232](#)
 - managing [225](#)
 - name [231](#)
 - operating [226](#)
 - VM group [179](#)
- CPU quota [138](#), [142](#), [238](#), [251](#), [452](#), [465](#), [484](#), [499](#)
 - for a VM group [178](#)
 - for monitor VM [56](#)
 - member VM [179](#)
 - of a VM [138](#), [452](#)
- CPU reconfiguration [211](#), [218](#)
- CPU share [641](#)
 - current [642](#)
 - effective [642](#)
 - planning [240](#)
 - recommendation [241](#)
- CPU time consumed [251](#), [391](#), [608](#)
- CPU utilization [138](#), [142](#), [176](#), [178](#), [452](#), [465](#), [499](#)
- CPU-POOL-NAME (operand) [443](#), [455](#), [456](#), [464](#), [466](#), [470](#), [473](#), [535](#), [549](#), [561](#), [574](#), [583](#), [610](#)
- CPU-QUOTA (operand) [438](#), [440](#), [455](#), [456](#), [465](#), [486](#), [499](#), [514](#)
- CREATE-VM [138](#), [154](#), [179](#), [215](#), [232](#), [452](#)
 - for the monitor VM [56](#), [72](#), [138](#), [452](#)
- CREATE-VM-ASSIGNMENT-SET [463](#)
- CREATE-VM-CPU-POOL [231](#), [464](#)
- CREATE-VM-GROUP [154](#), [178](#), [232](#), [465](#)
- creating
 - assignment set [463](#)
 - CPU pool [231](#), [464](#)
 - VM group [178](#), [465](#)
- current CPU share [642](#)
- D**
- DAMP [412](#)
- data center [20](#)
- data spaces [265](#)
- data type
 - S variables [425](#)
- deallocating devices from a VM [509](#)
- dedicated CPU [147](#), [249](#), [459](#), [488](#)
- DEFINE-CONSOLE [105](#), [108](#)
- DELETE-VM [84](#), [174](#), [180](#), [232](#), [467](#)
- DELETE-VM-ASSIGNMENT-SET [469](#)
- DELETE-VM-CPU-POOL [233](#), [470](#)
- DELETE-VM-GROUP [181](#), [232](#), [471](#)
- deleting
 - assignment set [469](#)
 - CPU pool [470](#)
 - VM group [471](#)
- DETACH-DEVICE [198](#), [219](#), [262](#)
- DETACH-VM-RESOURCES [218](#), [472](#)
- detaching real CPUs [472](#)
- device [359](#)
 - accounting [286](#)
 - assigning attribute [491](#)
 - BS2000 [34](#), [47](#)
 - change configuration properties [201](#)
 - changing usage mode of [495](#)
 - emulated [34](#), [47](#)
 - example [294](#)
 - exclusive use [292](#)
 - outputting information [556](#)
 - shared use [292](#)
 - snap unit [336](#)
 - the monitor VM [285](#)
- device assignment [285](#)
 - explicit [287](#)
 - implicit [194](#), [288](#)
- device name [285](#)
- device name, mnemonic [285](#)
- device peripherals [45](#), [46](#), [47](#), [284](#)
- device range
 - specifying [429](#), [493](#), [496](#), [510](#), [558](#), [581](#), [636](#)
- device release [285](#)
- device type code, unknown [46](#), [47](#)

devices

- allocate to a VM [428](#)
- deallocate from a VM [509](#)
- deallocate from one VM and allocate to another [635](#)
- diagnostic IPL [157](#)
- dialog startup [59](#), [69](#), [70](#)
- domain [30](#)
 - BS2000 [68](#)
 - Linux, Windows [30](#)
 - MONITOR [30](#), [68](#), [79](#)
- domain 0 [30](#)
- DRV [318](#)
- dual mode (DRV) [318](#)
- Dual Recording by Volume [318](#)
- dump testament [413](#)
- dynamic I/O configuration change
 - channel devices [361](#)
 - devices on X2000 [362](#)
- dynamic I/O configuration modification [361](#)
- dynamic PAV [348](#)
- dynamic reconfiguration of main memory [270](#)
- dynamic reconnection feature [342](#)

E

- effective CPU share [642](#)
- EMDS [106](#)
- End-Tag [367](#)
- END-VM-DIALOG [92](#), [115](#), [119](#), [167](#), [476](#)
- enlarging main memory for a VM [271](#), [478](#)
- error [423](#)
 - during VM2000 startup [395](#)
 - in monitor [397](#)
 - in the hypervisor [395](#)
 - in the VM2000 Agent [394](#)
 - in the Xen hypervisor [394](#)
 - in VMCONS [397](#)
- error class [423](#)
- error logging system [388](#)
- error types in VM2000 [393](#)
- errors
 - in the guest system [398](#)
 - in the monitor system [399](#)
- ETERNUS DX [336](#)

example

- \$VMCONS [124](#)
- ADMIN dialog from a user task [95](#)
- managing CPU pool [233](#)
- managing CPUs [253](#)
- MONJV [152](#), [274](#)
- operating VM groups [182](#)
- PAV [349](#)
- peripheral device management [294](#)
- PROP-XT [133](#)
- pubset management [324](#)
- shutdown [81](#)
- startup [73](#)
- exclusive device [292](#)
- executing
 - shutdown in the guest system [170](#), [623](#)
 - shutdown of VM2000 [79](#)
- explicit device assignment [287](#)
- explicit pubset assignment [307](#)
- EXTEND-VM-MEMORY [478](#)
- extra CPU [222](#), [442](#), [472](#)

F

- failure of a real CPU [218](#)
- FAST-STARTUP [59](#), [69](#)
- file \$TSOS.SYS.VM.SHUTDOWN [170](#), [625](#)
- firmware for VM2000 [28](#)
- foreign pool [34](#)
- functions of VMCONS [110](#)

G

- gatekeeper device [319](#)
- generation
 - for SCP [45](#)
 - hardware [45](#)
 - virtual console [45](#)
- global HOLD-VM [163](#)
- global interrupts [29](#)
- global RESUME-VM [165](#)
- global storage [277](#), [433](#), [516](#)
 - accounting [281](#)
 - assign [433](#)
 - cancel assignment [516](#)
 - changing attributes [280](#)

- global storage (continued)
 - changing size of 280
 - complex 282
 - exclusive use 277
 - partition 277
 - real 277
 - reconfiguration 281
 - shared use 277
 - takeover 281
 - virtual 196, 277
- GS complex 282
- GS prefix 120, 447
- GS server 282
- guaranteed messages 424
- guest system 23, 24, 642
 - and VM2000 194
 - constraints 40
 - initializing 84
 - installing 61
 - operating 84
 - shutdown 170, 623
 - terminating 84
- H**
- HA cluster 376, 377, 556, 579
 - conflict 377
 - Live Migration 378
 - takeover of a VM 377
 - VM name 377
 - VM2000 information 377
- hardware constraints 39
- hardware error logging (HEL) 211, 388
- hardware generation 45
- HEL file 388
- HELP-MSG-INFORMATION 423
- high availability 220
- high-availability cluster 376
- HIPLEX MSCF 338
- HNC 352
- HOLD-VM 163, 482
- home pubset 50, 52, 58, 59, 60, 111, 202, 289, 306, 309
- hot spare CPU 220
- HSI 25, 642
 - of the BS2000/OSD server 25
 - of the business server 25
 - virtual 25
- hypervisor 24, 28, 642
 - VM2000 24
 - Xen 24, 401
- hypervisor file 44, 69, 71
- hypervisor mode 29, 642
- hypervisor REP file 69, 71
 - installation 52
- hyphen as continuation character 449
- I**
- I/O configuration 46, 47, 196, 361
- identification VM-ID 139, 450
- IMON 43
- implementation of VMCONS 110
- implicit device assignment 288
- implicit pubset assignment 308
- INCLUDE-DEVICE-CONNECTION 199, 262
- information
 - on assignment sets 160
 - on CPU pools 160, 232
 - on devices 159
 - on load 160
 - on resources 159
 - on the VM status 160
 - on VM groups 160
 - on VM2000 194
 - services 159
- initializing
 - a guest system 84
 - a VM 138, 452
- installation monitor 43
- installing
 - the configuration file 56
 - the guest systems 61
 - the hypervisor REP file 52
 - the message file 51
- integer (data type of the S variables) 425
- interruption-free time switching 382
- i**

nterrupts

global 29

local 29

IO limit 360

IO utilization 138, 142, 360, 452

IO-PRIORITY (operand) 458, 488

IO-RESET (operand) 458, 487

IOCF file 45, 46

IOFCOPY 39, 45

IOLVM 360

IORM 348, 359

IPL 65

J

job variable for \$VMCONS 111

job variables 148, 272

K

KVP (console distribution program) 102, 107

L

Live Migration 136, 363, 376, 378

restrictions 378

VM2000 information 379

load library for VM2000 monitor 43

load library for VMDUMP 44

local HEL file 389

local interrupts 29

LOCATION (operand) 457, 503

location of VM1 in main memory 57

logging under VM2000 93, 116

logical console 84, 101, 119

M

machine check 221

macro VMGINF 194

main memory

accounting 265

assigning 268, 269

enlarging 271, 478

management 265

minimum size 140

reconfiguring 270

reducing 271, 506

main memory size of a VM 138, 140, 270, 452

main memory size of monitor VM 56, 266

malfunction alert 221

management

CPU pools 225

of global storage 277

of main memory 265

of pupsets 305

peripheral devices 211

pubsets 211

resources 35

MAX-CPU-UTILIZATION 239, 251

for monitor VM 56

MAX-CPU-UTILIZATION (operand) 455, 465, 499

MAX-IO-UTILIZATION

for the monitor VM 56

MAX-IO-UTILIZATION (operand) 455

MAXIMUM (multiprocessor level) 456

maximum CPU utilization 142, 176, 240

maximum IO utilization 142

meaning 423

measurement

with openSM2 392

with VM2000 391

member CPU quota 179, 452

memory management 211

memory space, contiguous 268

message code 51, 423

message file

for VM2000 69

for VM2000 monitor 43

for VMDUMP 44

installing 51

message flow

operating the guest system 121

message tables 118, 122

messages

asynchronous 93

from the VM2000 hypervisor 51

from the VM2000 monitor 51

guaranteed 424

in the case of STARTUP 51

- minimum main memory size
 - monitor VM 56
 - VM 140
- mirror pubset 318
- mnemonic device name 287
- MODIFY-IO-UNIT 201
- MODIFY-SYSTEM-PARAMETERS 262
- MODIFY-VM-ATTRIBUTES 161, 180, 484
- MODIFY-VM-DEVICE-ATTRIBUTES 491
- MODIFY-VM-DEVICE-USAGE 495
- MODIFY-VM-GROUP-ATTRIBUTES 180, 499
- MODIFY-VMCONS-PARAMETERS 167, 501
- modifying
 - attribute of a device 491
 - attributes of a VM group 499
 - VM attributes 161, 484
 - VMCONS parameters 167, 501
- monitor 23, 642
- monitor system 23, 642
 - configuring 48
 - deadlock 400
 - initializing 69
 - shutdown 79, 80
 - starting 68
 - terminating 83
- monitor VM 23, 148, 642
 - assigning attributes 56
- monitored data of VM2000
 - outputting 160, 391, 607
- MONJV 148, 272
 - beginning and end 148, 272
 - display 151, 274
 - example 152, 274
 - location 148, 149, 272
 - protection 148, 272
 - system part 150, 273
 - terminating the system abnormally 151, 274
 - user part 150, 273
 - validity 148, 272
 - values 150, 273
- MONO (multiprocessor level) 456
- mono mode (DRV) 318
- MOVE-VM 166, 503
- moving a VM 166, 503
- MSCF 338
- MSCF-COORDINATED (operand) 504
- MSGFIL 51
- MSGNOFL 51
- multiprocessor level 138, 452, 642
 - for a VM 143, 215
 - of monitor VM 57, 144, 215
- multiprocessor network 338
- N**
- name
 - assignment set 463, 469, 529
 - CPU ool 464
 - VM 454
 - VM group 465
- name components of the S variables 425
- network connection 352
- non-S procedures 91
- normal CPU 220
- normal unit 319
- notational conventions 18
- O**
- OCTO (multiprocessor level) 456
- OMNIS 101, 112
 - connection name 114
- openSM2 195, 392
- operating 89
 - a CPU pool 226
 - a guest system 84, 89
 - a guest system via \$VMCONS 117
 - a guest system via a virtual console 117, 336
 - a VM 89, 136
 - a VM in the ADMIN dialog 92, 115, 118
 - the VM2000 as a whole 88, 186
- OPERATING (privilege) 92
- operating concept of VM2000 88
- operating levels
 - combining (\$VMCONS) 114
 - of VM2000 88
 - splitting (\$VMCONS) 113
- operating mode 30, 68
- operation
 - of a VM group 177

- operator ID for \$VMCONS 60
- operator role for \$VMCONS 61
- OPERID password 167, 501
- OPNCON command 112
- original unit 319
- OSD/XC 25, 642
- output
 - assignment set 529
 - attributes of a CPU pool 160, 549
 - attributes of a VM group 160, 573
 - device information 159, 556
 - in S variables 425
 - monitored data of VM2000 160, 607
 - status of VM2000 resources 159, 579
 - VM attributes 532
- overdimensioned VM 215, 218, 224, 228, 229

P

- Parallel Access Volume 345
- parallel HIPLEX 283
- parameter file, automatic selection 55
- parameter service 39, 53
- parameter set OPR 105, 108
- PARMS field 66, 629
- pass message to VM administrator 527
- password
 - dialog access 57, 138, 144, 446, 452, 487, 501, 502
 - for MONJV 460, 480, 507
 - for UCON 167, 501
 - modifying 484
 - of VMCONS 167, 501
- password protection 211
 - in a VM 380
 - VMCONS 110, 380
- path 359
- path group ID 342, 587
- PAV (Parallel Access Volume) 345
 - dynamic 348
 - example 349
 - static 347
- performance 22
- pool pubset 321

- privilege
 - OPERATING 84, 92
 - VIRTUAL-MACHINE-ADMINISTRATION 91
 - VM2000-ADMINISTRATION 91
- privilege concept (SRPM) 91, 196
- privileges of the VM 138, 145, 161, 452, 484
- procedure file 449
- processor (CPU) management 211
- PROCESSOR (operand) 456
- program file
 - for VM2000 hypervisor 44
 - for VMDUMP 44
- pubset
 - accounting 306
 - addressing 305
 - devices 305
 - example 324
 - exclusive use 311
 - extend 321
 - managing 305
 - mirror pubset 318
 - pool pubset 321
 - saturation 321
 - shared use 311
 - SPACEPRO 321
- pubset assignment
 - explicit 307
 - implicit 308
- pubset device 305
- pubset name 305
- pubset reconfiguration 323

Q

- QUADRO (multiprocessor level) 456
- quick startup 59
- QUIET state 170

R

- Readme file 16
- real CPU 213
 - adding to a CPU pool 225
 - attaching 442
 - detaching 472
 - reconfiguring 218

- real CPU (continued)
 - status 214
 - real HSI 25, 643
 - reconfiguration
 - channel 263
 - hardware units 196, 262
 - paths 262
 - real CPU 218
 - virtual CPU 219
 - reconfiguring
 - main memory 270
 - REDUCE-PAGING-AREA 280
 - REDUCE-VM-MEMORY 506
 - reducing main memory for a VM 271, 506
 - redundant configuration 45
 - release item 43
 - release unit 43
 - REMARK 449
 - Remote service 102, 109
 - REMOVE-DEVICE-CONNECTION 200, 262
 - REMOVE-VM-DEVICES 509
 - REMOVE-VM-FROM-VM-GROUP 154, 180, 514
 - REMOVE-VM-RESOURCES 516
 - removing global storage 516
 - removing VM from VM group 514
 - REP file
 - for BS2000 69
 - for VM2000 69
 - for VM2000 hypervisor 69
 - RESET-VM 168, 519, 523
 - resource management 35, 161
 - resources 359
 - restart 83, 151, 398, 401
 - monitor system 403
 - restart routines of VM2000 401
 - RESTORE-VM-CONFIGURATION 364, 521
 - RMS file
 - for VM2000 hypervisor 44
 - for VM2000 monitor 43
 - for VMDUMP 44
 - RMS-Liefermenge 44
 - RUN file 130
 - running virtual CPUs on real CPUs 217
- S**
- S procedures 91, 130
 - S variables 159, 425
 - contents 425
 - data type 425
 - name components 425
 - saturation status 321
 - SAVE-VM-CONFIGURATION 363, 525
 - scheduling 217, 249, 643
 - fixed CPU assignment 249
 - time slicing 249
 - SCP 3970 102
 - SDF 91
 - SDF syntax file for VM2000 monitor 43
 - SDF-P 130, 159, 424, 425
 - selective HOLD-VM 163
 - selective RESUME-VM 165
 - self-loading BS2000 systems 36
 - SEND-VM-MESSAGE 166, 527
 - server unit
 - terminating 79
 - Service Level Agreements 20
 - SET-LOGON-PARAMETERS 92
 - SET-NAME (operand) 463, 469, 529
 - SETS
 - in the guest system 398
 - in the monitor system 399
 - monitor system 401
 - restart 401
 - shadow memory 268
 - shared device 338, 430
 - shared use 430
 - shared use of disks 338
 - SHOW-DEVICE-CONFIGURATION 194, 206, 262
 - SHOW-SYSTEM-INFORMATION 194, 202
 - SHOW-VM-ASSIGNMENT-SET 529
 - SHOW-VM-ATTRIBUTES 160, 532
 - SHOW-VM-CPU-POOL 160, 232, 549
 - SHOW-VM-DEVICE-STATUS 159, 556
 - SHOW-VM-GROUP 160, 181, 573
 - SHOW-VM-RESOURCES 159, 579
 - SHOW-VM-STATUS 160, 391, 607
 - SHUTARCH 172

- SHUTCTL [171](#), [624](#)
- shutdown [79](#), [80](#), [84](#), [170](#), [174](#), [623](#)
 - coordinated [81](#), [171](#)
 - example [81](#)
 - for the monitor system [83](#)
 - guest system [170](#)
 - QUIET state [170](#)
 - VM2000 [79](#), [170](#), [623](#)
- SHUTDOWN-VM [623](#)
- SLED [36](#), [410](#)
- SLEDFILE [410](#)
- snap session [337](#)
- snap unit [336](#)
- Snapset [337](#)
- SoftBooks [18](#), [51](#)
- software constraints [39](#)
- space provisioning [321](#)
- SPACEPRO [321](#)
 - execution [322](#)
 - under VM2000 [321](#)
- SPACEPRO pubset [321](#)
- spare CPU [220](#)
- SPD operation [338](#)
- splitting the operating levels (\$VMCONS) [113](#)
- SPVS operation [338](#)
- SQ manager [27](#), [32](#), [87](#)
- SQ server
 - operating mode [68](#)
- SRDF [320](#)
- standard CPU pool [34](#), [225](#)
- standard file name
 - configuration file [56](#)
 - hypervisor files [52](#)
 - overview [43](#)
- start
 - ADMIN dialog [155](#)
 - ADMIN dialog or VC dialog [446](#)
 - guest system on a VM [156](#), [627](#)
- start procedure file [449](#)
- Start-Tag [367](#)
- START-VM [156](#), [627](#)
- startup [70](#)
 - example [73](#)
- state of VM
 - at shutdown [173](#)
 - at termination [174](#)
 - DOWN [136](#)
 - IN HOLD [136](#), [157](#)
 - INITIALIZED [136](#)
 - MIGRATING [136](#)
 - NOT INITIALIZED [136](#)
 - on guest system termination [168](#)
 - RUNNING [136](#), [157](#), [163](#)
- state, changing
 - by means of DELETE-VM [175](#)
 - by means of HOLD-VM [163](#)
 - by means of RESET-VM [169](#)
 - by means of RESUME-VM [163](#), [165](#)
 - by means of SHUTDOWN-VM [173](#)
 - by means of START-VM [158](#)
- statement DEFINE-CONSOLE [105](#), [108](#)
- static PAV [347](#)
- status
 - of a VM [138](#)
 - real CPU [214](#)
 - virtual CPU [216](#)
- status diagram of a VM [136](#)
- STEP [449](#)
- string (data type of the S variables) [425](#)
- structured S variables [91](#)
- subsystem
 - VM2-MON [43](#)
- subsystem catalog for VM2000 monitor [43](#)
- SWITCH-VM-CPU [232](#), [631](#)
- SWITCH-VM-DEVICES [635](#)
- Symmetrix [345](#)
 - BCV [319](#)
 - clone [319](#)
 - gatekeeper device [319](#)
 - SRDF [320](#)
- Symmetrix Remote Data Facility [320](#)
- system availability [21](#)
- system parameter
 - MSGFIL [51](#)
 - MSGNOFL [51](#)
 - SHUTCTL [171](#), [624](#)
 - VMGIORAL [262](#)

system reset 145, 157
 system task 48, 397, 405

T

target unit 320
 task termination, abnormal 401
 Teleservice 102, 106, 109
 terminate
 a guest system 168, 519
 terminate VM 163, 482
 terminating
 a guest system 84
 a VM 174, 467
 ADMIN dialog 167
 ADMIN dialog or VC dialog 476
 server unit 79
 The 122
 trace entries 161, 484
 trace functions, internal 409
 transferring information 148
 TRIPLE (multiprocessor level) 456

U

UCON task 101
 unknown device type code 46, 47
 usage mode
 of a device 292
 of a subset 311
 user task 84
 utility routine VMDUMP 24, 406

V

VC dialog 117, 643
 VIRTGSxx (reserved name) 279
 virtual console 336, 643
 assigning 430
 operating concept 117
 virtual CPU 643
 reconfiguration 219
 status 216
 virtual CPUs 143, 215
 virtual HSI 25, 643
 virtual machine 23, 643
 virtual machine system 643

VIRTUAL-MACHINE-ADMINISTRATION

(privilege) 91
 VM 643
 adding to a CPU pool 225
 VM administration 89
 VM administrator 89, 643
 monitor VM 136
 send message 166, 527
 VM administrator commands 89, 187, 416, 644
 privileged functions 188, 190
 VM global HEL file 389
 VM group 20, 644
 adding to a CPU pool 225
 CPU pool 179
 creating 178
 deleting 471
 example 182
 VM added 138, 142, 452
 VM identification 644
 VM index 29, 92, 139, 450, 457, 644
 for a VM 138, 452
 for monitor VM 57
 VM mode 29, 644
 VM name 139, 644
 as accounting ID 382
 for a VM 138, 452
 for monitor VM 56
 naming recommendations 139
 VM prefix 116, 447
 VM, overdimensioned 215, 218, 224, 228, 229
 VM-GROUP-NAME (operand) 436, 438, 465, 471, 499, 514
 VM-ID 139, 450
 VM1 645
 VM2-MON, subsystem 43
 VM2000 644
 areas of application 19
 attributes 21
 changes 17
 concepts 20
 constraints 39
 coordinated shutdown 80
 hypervisor 24
 in guest systems 194

- VM2000 (continued)
 - in the availability cluster 21
 - in the data center 20
 - initializing 65
 - monitor 23
 - shutdown 79, 170, 623
- VM2000 administration 88
- VM2000 administration application \$VMCONS,
see VMCONS
- VM2000 administrator 88, 136, 186, 644
- VM2000 administrator commands 187, 416, 644
- VM2000 agent 31, 393, 394
 - abnormal termination 401
 - restart 401
- VM2000 command 644
- VM2000 command description
 - ADD-VM-DEVICES 428
 - ADD-VM-RESOURCES 433
 - ADD-VM-TO-VM-GROUP 436
 - ASSIGN-VM-GROUP-TO-CPU-POOL 438
 - ASSIGN-VM-TO-CPU-POOL 440
 - ATTACH-VM-RESOURCES 442
 - BEGIN-VM-DIALOG 446
 - CALL-VM-PROCEDURE 449
 - CREATE-VM 452
 - CREATE-VM-ASSIGNMENT-SET 463
 - CREATE-VM-CPU-POOL 464
 - CREATE-VM-GROUP 465
 - DELETE-VM 467
 - DELETE-VM-ASSIGNMENT-SET 469
 - DELETE-VM-CPU-POOL 470
 - DELETE-VM-GROUP 471
 - DETACH-VM-RESOURCES 472
 - END-VM-DIALOG 476
 - EXTEND-VM-MEMORY 478
 - HOLD-VM 482
 - MODIFY-VM-ATTRIBUTES 484
 - MODIFY-VM-DEVICE-ATTRIBUTES 491
 - MODIFY-VM-DEVICE-USAGE 495
 - MODIFY-VM-GROUP-ATTRIBUTES 499
 - MODIFY-VMCONS-PARAMETERS 501
 - MOVE-VM 503
 - REDUCE-VM-MEMORY 506
 - REMOVE-VM-DEVICES 509
 - REMOVE-VM-FROM-VM-GROUP 514
 - REMOVE-VM-RESOURCES 516
 - RESET-VM 168, 519
 - RESTORE-VM-CONFIGURATION 521
 - RESUME-VM 523
 - SAVE-VM-CONFIGURATION 525
 - SEND-VM-MESSAGE 527
 - SHOW-VM-ASSIGNMENT-SET 160, 529
 - SHOW-VM-ATTRIBUTES 532
 - SHOW-VM-CPU-POOL 160, 549
 - SHOW-VM-DEVICE-STATUS 159, 556
 - SHOW-VM-GROUP 160, 573
 - SHOW-VM-RESOURCES 159, 579
 - SHOW-VM-STATUS 607
 - SHUTDOWN-VM 170, 623
 - START-VM 627
 - SWITCH-VM-CPU 631
 - SWITCH-VM-DEVICES 635
- VM2000 commands 187, 416, 419
 - extending the command range 192, 193
 - grouping according to function 419
 - restricting the command range 192
 - restrictions 188, 191
- VM2000 configuration 363
 - backup file 367
 - change after the backup 364
 - Components not backed up 371
 - restore 211, 364, 521
 - save 211, 363, 525
- VM2000 dump 24, 406
- VM2000 errors 395
- VM2000 HSI 28, 29, 644
- VM2000 hypervisor 53
 - abnormal termination 401
 - Restart 401
- VM2000 operation
 - terminating 79
- VM2000 parameter set 55
- VM2000 parameters 53
- VM2000 procedure file 130
- VM2000 virtual machine system 13
- VM2000-ADMINISTRATION (privilege) 91
- VM2C task 48, 397, 405
- VM2E task 48, 397, 405

VM2G task 48, 397, 405
VM2J task 48, 397, 405
VM2M task 48, 397, 405
VM2P task 48, 397, 405
VMCONS 24, 110, 446, 644
 clear down connection 477
 connection via OMNIS 112
 example 124
 functions 110
 implementation 110
 job variable 111
 operator ID 60
 operator role 61
 parameters 167
 password 110, 112, 380
 password in parameter file 112
 preparing to use 60
 VM2000 administration application 110
VMDUMP 406
 controlling trace functions 409
 separate program 406
 utility routine 24
VMGINF (macro) 194
VMGIORAL 262
VMHEL 389, 390
VMHEL commands 390

W

wait state of a VM 163
 cancel 163, 482, 523
 global 163
 selective 163
WAIT, state types 163
When 122

X

X2000 30, 34, 47, 62
Xen hypervisor 24, 27, 30, 394
XenVM 645
XML document 367

